ADAPTATIONS AND ANALYSIS OF THE AFIT NOISE RADAR NETWORK FOR INDOOR NAVIGATION

THESIS

Russell D. Wilson IV, Second Lieutenant, USAF

AFIT-ENG-13-M-50

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-13-M-50

ADAPTATIONS AND ANALYSIS OF THE AFIT NOISE RADAR NETWORK FOR INDOOR NAVIGATION

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Russell D. Wilson IV, B.S.E.E.

Second Lieutenant, USAF

March 2013

AFIT-ENG-13-M-50

ADAPTATIONS AND ANALYSIS OF THE AFIT NOISE RADAR NETWORK FOR INDOOR NAVIGATION

Russell D. Wilson IV, B.S.E.E.
Second Lieutenant, USAF

Approved:

Peter J. Collins, PhD (Chairman)	Date
John Raquet, PhD (Committee Member)	Date
Lt Col Geoffrey A. Akers, PhD (Committee Member)	Date

AFIT-ENG-13-M-50

Abstract

After several years of development, the AFIT Noise Radar Network (NoNET) has proven to be an extremely versatile system for many standard radar functions. This pallet of capabilities includes through the wall target tracking capabilities due to its wide bandwidth and UHF operations. Utilizing White Gaussian Noise as its waveform, the NoNET can operate at much lower power levels than other comparable systems while remaining extremely covert. In an effort to explore new applications, the question arose could the NoNET provide a viable option for navigation capability in GPS denied and indoor environments? This research aims to provide proof of concept and demonstration of the navigation function execution with the NoNET in indoor, multipath-ridden environments. Results demonstrate that the NoNET is currently capable of locating a receiver to accuracies of approximately 1 foot. Multipath, background RF interference, and network timing were investigated and solutions to mitigate the limitations imposed by each were developed with the potential to significantly improve accuracy. Future upgrades to the current NoNET hardware package were also investigated in order to provide a near real-time, portable solution to navigation in GPS denied environments.

To my Fiance, the love of my life.

Acknowledgments

First and foremost, thank you Dr. Collins for all that you have done for me during my stay at AFIT. Your knowledge will follow me for the rest of my career and beyond. I would also like to thank Mr. McNeely for his extensive assistance in construction and debugging of the hardware. His help proved invaluable to completion of this thesis. Special thanks to the AFIT ANT Center as well for your sponsorship and facility usage. Finally, I would like to thank each of the other 4ILY students in GE-13M. Your knowledge, experience, and support has provided me a solid foundation for the future.

Russell D. Wilson IV

Table of Contents

Page

Page

List of Figures

Figure Page

List of Tables

List of Acronyms

Acronym	Definition
ADC	analog to digital converter
AFIT	Air Force Institute of Technology
ANT	Advanced Navigation Technology
CW	continuous wave
DAC	digital to analog converter
DSP	digital signal processing
EM	electromagnetic
GPS	Global Positioning System
IC	integrated circuit
IEEE	Institute of Electrical and Electronics Engineers
INS	inertial navigation systems
LPA	log-periodic antenna
LPD	low-probability-of-detection
LPI	low-probability-of-intercept
MCU	micro-controller
NoNET	Noise Radar Network
NTP	Network time protocol
NTR	Noise technology radar
RF	radio frequency
RFI	radio frequency interference
SNR	signal-to-noise ratio
TDOA	time difference of arrival
TDR	Time-domain reflectometer

Acronym	Definition
UHF	ultra-high frequency
UWB	ultra-wideband

ADAPTATIONS AND ANALYSIS OF THE AFIT NOISE RADAR NETWORK FOR INDOOR NAVIGATION

I. Introduction

1.1 Problem Description

In today's society, it seems that almost everyone has become reliant upon an unseen network of technology. It is used by the majority of the population now, and provides a simple method of navigation for a limitless number of uses. Known as the Global Positioning System (GPS), its beginnings can be traced back to the 1960's. The GPS consists of a network of positioning satellites in Earth's orbit, which provides a method of global navigation to anyone having the appropriate receiver technology. Receivers were purposely designed to be simple and can be built for a fraction of the cost of other navigation solutions. This navigation technology has no upper limit to the number of users. Receivers are now integrated into automobiles, cameras, and even personal cell phones. The GPS has also become a powerful tool for military use as well, providing capabilities never before imagined in terms of vehicle and munitions guidance with pinpoint accuracy. A drawback to the GPS is once a receiver is brought indoors, it most often ceases to function. The GPS fails to provide service to various indoor environments, mainly due to the extremely low power signals being transmitted from Earth's orbit [6].

This issue complicates many modern day scenarios. For example, how does one find their location inside buildings or in other locations where a clear view of the sky is not readily available? Indoor navigation becomes particularly important during search and rescue operations. Emergency responders may require navigation assistance within complex structures. Many different techniques have been attempted and developed over the

past decade to address this problem with mixed results. For example, inertial navigation systems (INS) can be very accurate, but only for a short time period. They are prone to drifting when not paired with another form of correction. Recent research involves ultra-wideband (UWB) and lower frequency transmissions. These types of signals significantly increase the accuracy and ability for electromagnetic (EM) transmissions to propagate through walls, while still remaining useful to the user. These UWB properties have many different advantages, but still suffer from the fact that they operate mainly at short ranges in comparison to the global navigation solutions at similar power levels. Another useful property for navigation is rejection of external interference and jamming from other electronic devices [3, 4].

The question remains. How can we perform a navigation function in an indoor or cluttered environment in a simple manner, inexpensively?

1.2 Motivation

The Air Force Institute of Technology (AFIT) Noise Radar Network (NoNET) could provide a viable solution for indoor navigation. The NoNET consists of a network of individual Noise technology radar (NTR) units. Past research has utilized these UWB noise-based systems to perform imaging, target tracking, and target characterization through building walls [12]. NTR is a low frequency, baseband, software-defined system. The software can be easily modified from a sole radar platform to a system which could perform similarly to the GPS. By operating in the ultra-high frequency (UHF) band, NTR has the ability to operate through building walls and other obstacles. Radio stations also utilize these lower frequency ranges to reach consumers regardless of these obstacles.

Data transfer using the noise waveform is also undergoing research at AFIT. Potential uses include data transfer while simultaneously conducting radar measurements or performing various other functions. Miniaturization efforts are also taking place, which could allow greater portability while simultaneously reducing power consumption and cost.

This effort promises to provide real-time receiver processing capabilities. By operating in real-time, asynchronous communication and networking operation can be achieved. As a whole, this system could provide a multitude of functionalities in a wide range of scenarios, all while utilizing the same simplistic hardware and software package. Further details on the NTR hardware are provided in Chapter 2.

1.3 Goals & Assumptions

The primary purpose of this research is to provide proof of concept and the initial trial demonstrations of utilizing the AFIT NoNET to perform a navigation function in indoor environments. Specific performance parameters of the system as well as its advantages and disadvantages will be discussed. First, preliminary designs have shown that the AFIT NoNET will require minimal hardware modification, if any, in order to crudely perform this navigation function. Software, however, will require significant modifications in order to convert from the current radar operation mode to a navigation mode. Specifically, the traditional two-way monostatic radar ranging must be modified to a one-way time-delay ranging, and development of a method to remote control the network of radars utilizing MATLAB must occur. MATLAB is the preferred software development tool as all previous work with NTR was conducted with it. The remote control requirement allows for commands and data transfer across the network to be controlled from a central computer.

This system is bulky at this stage in its development and can require a significant amount of computational power to perform radar functions. But, the end result of this thesis will demonstrate and analyze the capabilities of such a versatile system rather than focusing on usability and portability.

One obstacle yet to be overcome with the NoNET is that of collection trigger synchronization. The NoNET currently lacks real-time receiver capability, so captures on each node, or NTR unit, must be synchronized with a trigger. Synchronization of these capture triggers is required in order to perform GPS-like navigation. This differs

from previous research, where the nodes across the network were not required to capture simultaneously in order to perform the networked function. This research utilized a hard-wire trigger connection to solve this timing problem. Realistically, this solution is cumbersome and inefficient for a fielded application. As a secondary objective for this project, this trigger timing issue was further investigated to locate another, potentially wireless, solution. A wireless triggering solution pushes the navigation functionality of the NoNET toward more practical applications.

Another aspect investigated was radio frequency (RF) multipath. When operating in cluttered environments, the multitude of RF energy reflections can confuse a simple correlation-based receiver such as NTR. A challenge and objective to this navigation system research will be the mitigation of these multipath degradations. In summary, the goals of this research include:

- Development of the software required to utilize the NoNET as a navigation tool
- Demonstration and a comprehensive analysis of the NoNET's navigation capabilities in an indoor environment
- Analysis and mitigation of indoor multipath effects
- Synchronization of capture triggers across the network
- Development of a method for wireless trigger synchronization

1.4 Background

The AFIT NoNET has been under development for several years through a series of master's theses. It was originally designed and constructed based on a system built at Pennsylvania State University for the use of through-wall imaging [12]. The NoNET is a series of NTR nodes, each of which can operate as independent radars, or as a cooperative

network, capable of producing bistatic/multistatic radar images. NTR is unique because it utilizes amplified random thermal noise as its transmission waveform.

Work with NTR began under Ashley Schmitt in 2009 with a focus on through-wall imaging [12]. Following Schmitt, Matthew Nelms' work continued the development of the system by implementing various improvements aimed to allow true bistatic and multistatic radar ranging measurements [10]. Following these projects, various other theses focused on waveform analysis, velocity/Doppler measurements, system simulations, and UWB antenna designs [8, 9, 13]. A picture of the NTR as it stood in September of 2012 is shown in Figure 1.1. Significant modifications were conducted to the primary hardware by Joshua Hardin in October of 2012.

1.5 Document Organization

This document is organized closely following the research process. Chapter 2 provides the theory, logic, and principles behind the research conducted. It is based upon a comprehensive literature search and mathematical analysis. The methodology utilized in the research process is described in Chapter 3. Chapter 4 provides detailed documentation and analysis of the research results. Finally, Chapter 5 forms the final conclusions and suggests future work for further development and integration of this research.

Figure 1.1: Picture of NTR as it stood in September 2012.

II. Theory & Noise Radar Overview

This chapter serves as an overview of the theory behind the AFIT NTR. The structure and operation of NTR is described. A minutia of the EM theory describing the various phenomenon found when operating RF devices in indoor environments is provided. Finally, the basic principles and mathematical derivation behind time difference of arrival (TDOA)-based navigation is outlined.

2.1 Noise Radar Theory

2.1.1 Continuous Random Noise Waveform.

NTR is a baseband radar platform which utilizes white-Gaussian electronic noise as its signal source. This differs from the traditional radar system, which utilizes a more deterministic waveform such as a sinusoid or other repetitive signal. Typically, these systems electronically mix these waveforms to higher frequencies. As a baseband system, NTR does not mix its waveform but directly transmits into the environment with no purposeful frequency modifications. Figure 2.1 shows an example of the difference between a traditional sinusoid radar waveform and a noise waveform. Due to the statistical orthogonality between each sample of the noise waveform, correlation can detect this transmission even in extremely low signal-to-noise ratio (SNR) situations where correlation with a more traditional waveform is less effective.

NTR also differs from most traditional radars in that it is UWB. Its operational frequency range spans from approximately 300 MHz up to approximately 1 GHz, primarily limited by the antennas. This operating frequency range qualifies NTR as an UWB communication/radar system per the Institute of Electrical and Electronics Engineers (IEEE) guidelines on UWB technology. This qualification is based on a system's fractional

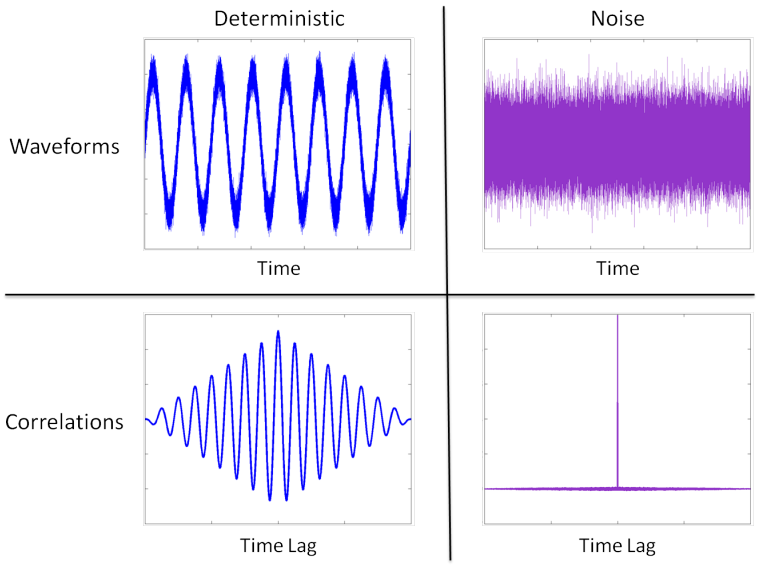


Figure 2.1: Demonstration of the correlation advantage of the noise waveform verses a traditional radar waveform.

bandwidth given by

$$B_f = \frac{2\,(f_h - f_l)}{f_h + f_l}. \tag{2.1}$$

Per this IEEE guideline, systems greater than 25% fractional bandwidth qualify as UWB. The AFIT NTR currently operates with a fractional bandwidth of approximately 100% [1]. This UWB property has distinct advantages because it significantly improves a radar's range resolution. In other words, increased bandwidth narrows the correlation peaks in Figure 2.1 more than a traditional narrow bandwidth operation. A distinct disadvantage to such a wide bandwidth is that Doppler, or velocity, estimation of a target cannot be directly measured and must be determined by different, highly computationally rigorous techniques [13].

The AFIT NTR is categorized as a continuous wave (CW) radar. Being CW, the NTR does not pulse its output, but rather continuously transmits a signal. This transmit waveform is in contrast to the traditional pulsed RF waveform of most operational radars. Those

systems transmit a short pulse, then transition to a receive mode to obtain return reflections of that pulse. This CW property provides distinct advantages in terms of the system's low-probability-of-detection (LPD) and low-probability-of-intercept (LPI) properties. In other words, NTR can operate in an environment with little chance of detection by a noncooperative receiver. At an elevated level, any external capture of this waveform will appear as nothing more than ambient noise. Finally, by operating in a CW mode, radar range ambiguity is also eliminated [11].

2.1.2 Noise Radar Hardware Construction.

2.1.2.1 Pre-September 2012 Platform & Signal Flow.

NTR's fundamental design is laid out in Figure 2.2. The hardware pre-September 2012 followed this design.

The overall construction of NTR is simple in comparison to other CW radars. The system begins with a commercial thermal noise source, which produces a uniform frequency response from DC to approximately 1.2 GHz. This noise source is then filtered to approximately 400-750 MHz. Next, the transmit signal is split before transmission to provide a reference transmit signal to an on-board analog to digital converter (ADC) for later digital signal processing (DSP) in MATLAB and amplified before being transmitted through the antenna into the environment.

The next step in the functional chain of the AFIT NTR is the receive path. This system utilizes a direct conversion receiver, unlike the traditional heterodyne receivers found in many radars. All filtering, sampling, and analysis are performed at baseband. The signal captured by the receive antenna is amplified, then filtered through a band-pass filter to help improve the receive SNR. Next, the captured waveforms are converted to digital through a second channel on the ADC for processing.

The current ADC for NTR operates on both of its channels at a maximum rate of 1.5*G*samples per second. This data is collected and dumped into a MATLAB workspace on a connected computer. This computer manages the radar while also performing all DSP.

2.1.2.2 Post-September 2012 Hardin Modifications.

At the beginning of this work, it was determined that extra capability needed to be included into NTR to allow for greater flexibility in future work. Previous projects suggested that synthetic generation of the noise waveform would be extremely useful for communication and simplification of the multistatic capture process. Joshua Hardin, also

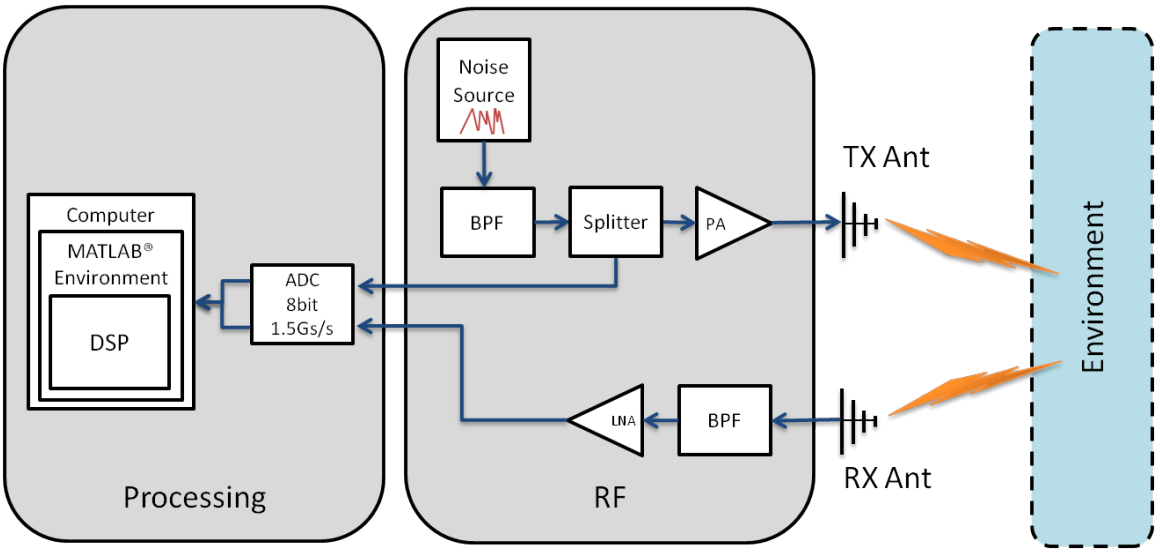


Figure 2.2: Structure diagram of the September 2012 AFIT NTR.

a Master's student of AFIT's GE-13M class, was the primary developer for this hardware upgrade. Figure 2.3 outlines the new system structure. Hardin added the capability to utilize a digital to analog converter (DAC) card and various other electronic switches. These switches allow for software selection between this synthetic waveform generator and the analog noise source. These new additions also added the capability to toggle use of the bandpass filters and to dial in the desired level of both transmit and receive attenuation.

This proved extremely vital in multistatic or multiple node operations where signal power levels due to the high number of transmitters operating in a small area were saturating the capture card input channels. These additions also allowed the system to not only transmit noise, but any waveforms which could be synthesized and uploaded to the DAC [5].

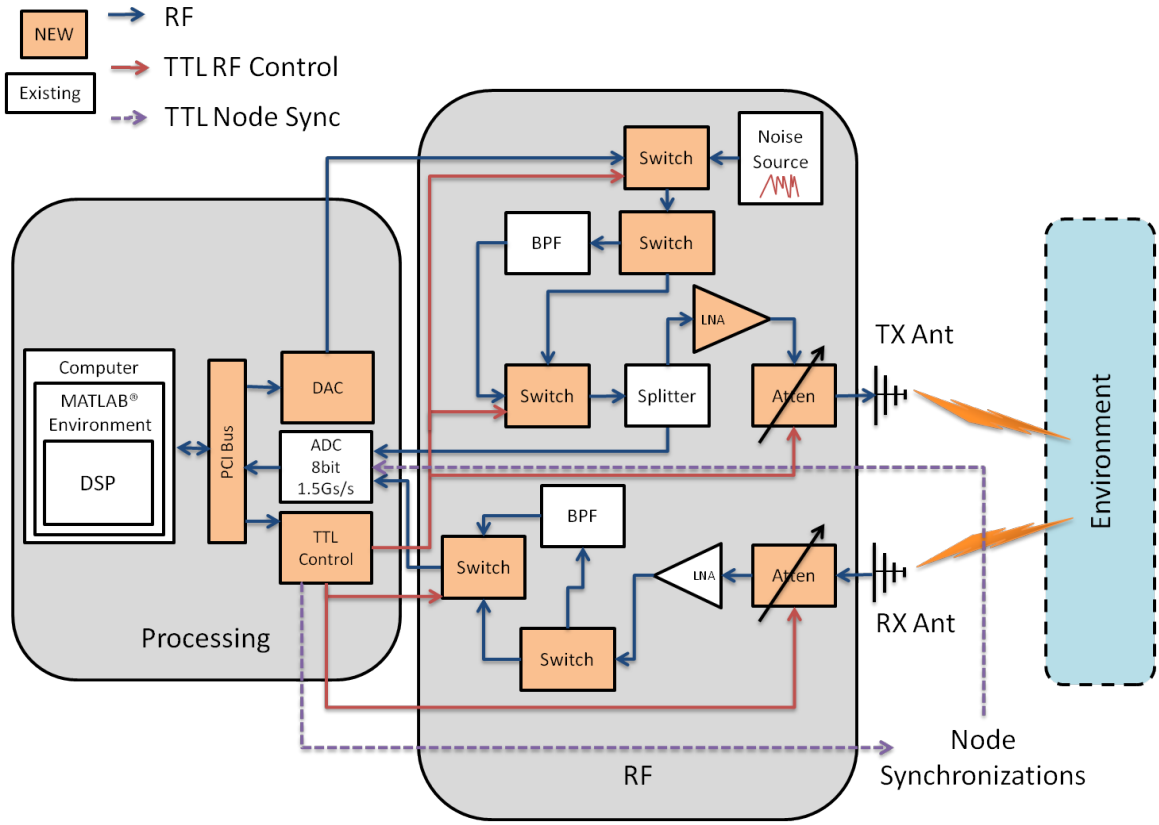


Figure 2.3: Joshua Hardin's modifications to the September 2012 AFIT NTR.

2.1.2.3 NTR Antennas.

Omnidirectional antennas were desired for simple through-the-wall ranging situations and will be required for using NTR as a navigation beacon. Highly directional antennas would not allow transmissions from each NTR node to all other nodes on the network independent of their placement and orientations. A unique challenge arises when trying to

select an antenna for use with NTR due to its large bandwidth. Most antennas only perform well in very narrow bandwidths. The original antenna NTR was designed with log-periodic antenna (LPA)s similar to that pictured in Figure 2.4 [12].

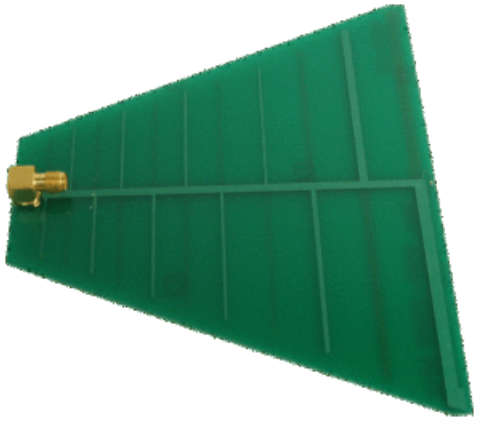


Figure 2.4: Picture of the LPA utilized on the AFIT NTR.

In the past, arrangement of these antennas was experimented with due to complications involving the mutual coupling between the transmit and receive antennas. Due to the close proximity of the two antennas, certain alignments produced larger amounts of coupling. An example of this problematic alignment is placing the antennas in the same plane of elevation and orienting both antennas for a vertical waveform polarization as is demonstrated in Figure 1.1. The current NoNET configuration calls for the antennas to be stacked vertically and oriented for vertical polarization in order to mitigate this coupling complication.

2.1.2.4 NoNET Networking and ADC Capture Triggers.

Each NTR computer can be linked together over a wireless or wired network to construct the NoNET. With multiple nodes, functionality has been shown to produce a "netted monostatic," bistatic, and even potential for multistatic operation modes. These modes can construct radar images from all NTR nodes in the NoNET to obtain

multidimensional radar images of the sampled environment. The bistatic and multistatic modes have not been fully developed and utilized to date. The reason for this lack of utilization is that a triggering scheme has yet to be devised to synchronize all of the involved nodes in their digital capture routines [10, 12]. This is why some of the work for this project will focus on methods by which simultaneous capture triggering of all nodes on the network can be accomplished wirelessly at greater distances.

There are two ways by which this triggering synchronization can be obtained: synchronizing the actual issued triggers (for example, exactly equal cable lengths to each node), or measuring the trigger synchronization error in some fashion and accounting for it in post-processing. For this navigation process, a combination of these two approaches will be utilized. The required trigger synchronization accuracy only needs to exceed that of NTR's temporal resolution. With an ADC sampling rate of 1.5G samples per second, NTR's resolution is approximately 0.7ns, equating to approximately 0.2m or 0.66ft of electrical distance in free space. Thus, in order to obtain perfect capture synchronization across the network utilizing cables, each of the cable lengths must be within this electrical distance of one another. Any other method must synchronize within the 0.7ns threshold to be considered perfectly synchronized.

Each ADC has multiple triggering inputs, which can be utilized to initiate data capture. These options include: software generated, an external BNC connection, a TTL SMC connection, and even the ability to trigger off of encoder inputs. Past NoNET work utilized the external BNC trigger connection with a separate function generator [10]. For this project, nodes will be placed at much greater separation distances, making the use of long lengths of coaxial cable impractical at best. Attempts were made to utilize both the software trigger and the SMC header trigger of these capture cards for this project.

2.2 Through-Wall EM Transmissions

Through wall transmission radar has become a rapidly growing field of study for the UWB community. Simply put, there are a variety of parameters which must be taken into account for this sort of application to function well. These parameters include the attenuation effects of the wall materials, the multilayer effects an EM signal will experience due to the changes in propagation medium, the operation frequency, the angle of incidence to the structure, and the polarization utilized [7]. This research assumes the walls are not primarily constructed from metal due to the fact that they would be extremely difficult, or even impossible, to transmit through. For experiments conducted at AFIT, the majority of the indoor walls are constructed of concrete.

2.2.1 Wall Attenuation.

Since it is assumed most walls are constructed from wood and/or concrete, it is necessary to analyze the losses the EM waves encounter when penetrating the wall. Table 2.1 is a list of common materials, their relative dielectric constants, ϵ_r, and their conductivity, σ_c.

Using the equation for the attenuation constant, α,

$$\alpha = \omega \sqrt{\mu \epsilon_r} \left\{ \frac{1}{2} \left[\sqrt{1 + (\frac{\sigma_c}{\omega \epsilon_r})^2} - 1 \right] \right\}^{\frac{1}{2}} \tag{2.2}$$

and the fact that

$$Attenuation = 20 \log_{10} (e^{-\alpha z}) \tag{2.3}$$

$$= 20(-\alpha z) \log_{10}(e) \tag{2.4}$$

$$= -8.68(\alpha z) \; in \; dB/m \tag{2.5}$$

the attenuations for the different materials per meter of propagation can be estimated at a fixed frequency (Table 2.1) [7]. From these values, it can be assumed that most of the loss of RF energy comes from the multiple reflections through the multilayer wall [12].

Table 2.1: Material properties [7]

Material	ϵ_r	$\sigma_c(mS/m)$	α @500 MHz
Air	1	0	0
Metal(iron)	1	10^8	55,774
Fresh Water	80	1	~ 0
Sea Water	81	4×10^3	0.25
Dry Soil	2-6	$10^{-1} - 1$	~ 0
Dry Concrete	6	1	~ 0
Dry Clay	3	1-10	~ 0

2.2.2 Multilayer Model.

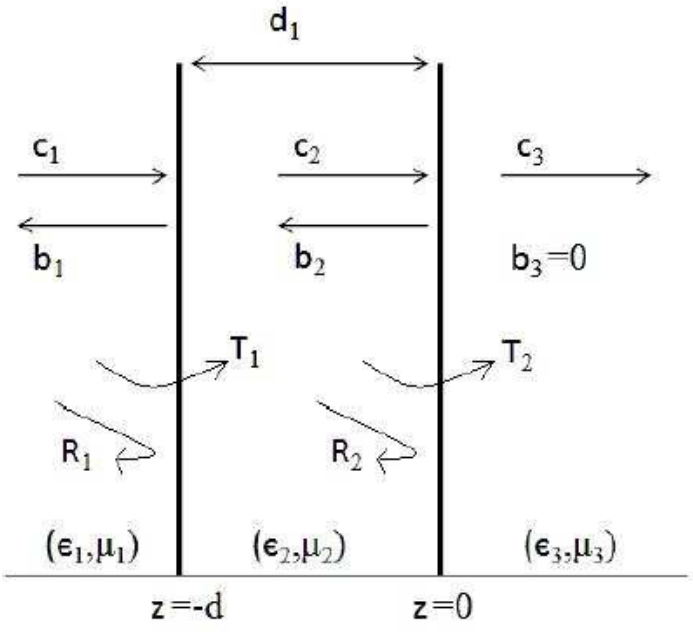


Figure 2.5: Multi-layer model of wave propagation through walls [2].

Figure 2.5 depicts transmission and reflection coefficients for a wave traveling through a wall, where c_1 is the forward wave amplitude, and b_1 is the reverse. The relationship is given by the following equation [2]:

$$[c_1\ b_1]^T = \Pi_{n=1}^{2}\left[A^{(n)}\right][c_3\ b_3]^T \tag{2.6}$$

$$\left[A^{(n)}\right] = \frac{1}{T_n}\begin{bmatrix} e^{jk_{nz}d_n} & R_n e^{-jk_{nz}d_n} \\ R_n e^{jk_{nz}d_n} & e^{-jk_{nz}d_n} \end{bmatrix} \tag{2.7}$$

where

$$R_n = \frac{Z_n - Z_{n-1}}{Z_n + Z_{n-1}}\ ,\ T_n = 1 + R_n \tag{2.8}$$

$$Z_n = \frac{\eta_n k_n}{k_{nz}}\ ,\ k_{nz} = \sqrt{k_n^2 - k_z^2}\ ,\ k_z = k_o sin(\theta_i). \tag{2.9}$$

and θ_i in this case is the angle of incidence. This relationship shows that the angle of incidence plays a key role in the propagation losses, and that perpendicular transmission incidence angles are preferred in order to obtain the greatest transmitted amplitude. It is assumed the polarization does not change upon interaction with the wall. Multipath effects beyond the wall are also neglected in this case. Research in the through-wall interactions with NTR was conducted by Lai at Pennsylvania State University in 2007 [7]. For simulation purposes in Lai's work, the following relationships were gathered as well:

$$\begin{aligned} \bar{E}^i &= \hat{x}E_o e^{-jk_z z} \\ \bar{E}^r &= \hat{x}\Gamma E_o e^{-jk_z z} \\ \bar{E}^t &= \hat{x}T E_o e^{-jk_z z} \\ \Gamma &= \frac{b_1}{c_1}\ ,\ T = 1 + \Gamma \end{aligned} \tag{2.10}$$

Lai's simulations aimed to understand frequency selection impacts and the multilayer phase impacts of the wall. The first simulation involved the wall losses over different frequency bands. The frequency band between 450 and 750 MHz had the lowest and most consistent amount of attenuation loss with respect to frequency. This simulation assumed a concrete, 10 cm thick wall. Higher frequencies degraded through wall transmission performance.

Lai also conducted hardware experiments with a network analyzer to verify his simulation results. The real world results in the band of 450-750 MHz (around the NTR operating range) followed simulation results very closely. Thus, by selecting the proper frequencies for the thickness and wall materials provided, attenuation effects can be estimated from direct path losses alone with minimal error [7].

2.3 Navigation Fundamentals

The last portion of this chapter involves a brief overview of the GPS and how the principles in use for this technology were implemented into this thesis research. The GPS can be thought of as a beacon-based navigation system. A beacon-based navigation system refers to a system constructed from a series of transmitters. Each transmitter sends out a reference signal that the receiver utilizes to determine the range, ρ_n, between itself and that transmitter (Figure 2.6). The navigation receiver can then utilize a series of these range measurements, in conjunction with the known transmitter locations, to calculate its location in space. Solutions to the navigation problem require at least three transmitter locations, as well as, the range measurements to each of those transmitters in a perfect environment. More than three transmitter range measurements can significantly improve location accuracies when error is present within the system and are physically required in most situations where the unit clocks are not perfectly synchronized [6].

Ranges are determined from the received transmitter signals by dividing the speed of electromagnetic propagation by the wireless signal time lag. In an equation form, also

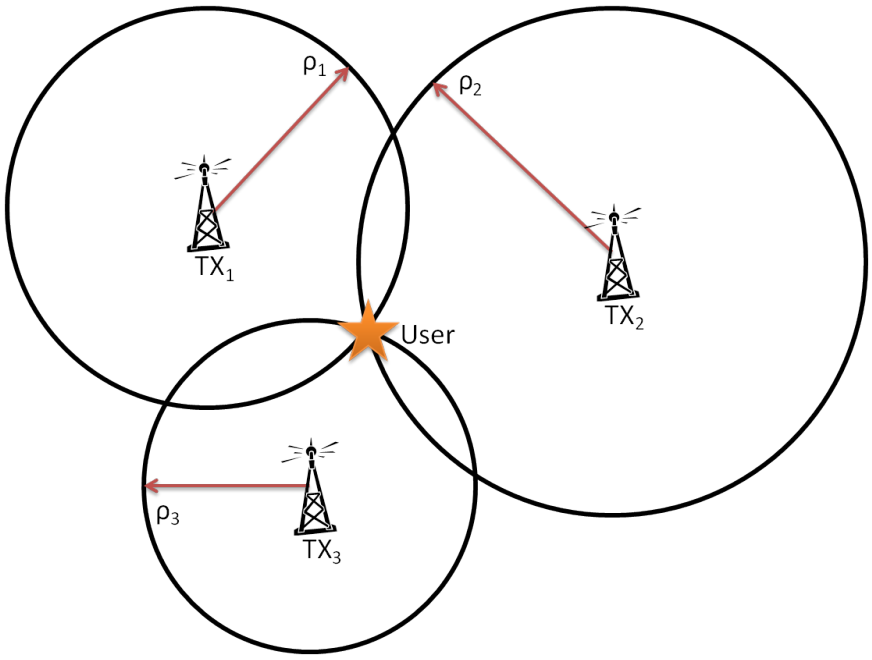


Figure 2.6: Example of beacon-based navigation.

known as the radar range equation to the radar community,

$$r = \frac{c\Delta t}{2}, \tag{2.11}$$

where r is the range of the return, c is the speed of light through free space, and Δt is the time delay between transmission and reception. In this case, the factor of one half arises from the fact that most monostatic radar ranges are measured as a two-way time delay. In radar, the signal interacts with the target at exactly half the propagation time. For one-way ranging measurements used in the GPS or any other beacon-based systems, this relationship simplifies to

$$r = c\Delta t. \tag{2.12}$$

This range is not typically accurate as many other factors introduce error and must be accounted for. One example of this is the atmospheric propagation changes in each layer of Earth's atmosphere for the GPS. Another factor is system clock errors between each

transmitter and receiver. Essentially, each time delay measurement will inherently include the clock error from the true system time for the transmitter and the receiver, as well as the true range propagation delay. The resulting measurement is known as a pseudorange and includes all these errors wrapped up into one value. In the case of NoNET, clock error is the trigger synchronization error.

2.3.1 TDOA Difference Measurements and Least Squares Algorithm.

Once the raw pseudoranges are captured, the process for extracting an estimated position for the receiver can begin. As mentioned before, these pseudoranges include the clock error between both the transmitter and receiver with reference to the true time. The transmitter clock error can be measured and accounted for. The receiver clock error remains an unknown.

Utilizing a process known as TDOA, the clock error between two transmitters can be measured. Figure 2.7, demonstrates this process. In this scenario, transmitters 2 and 4 can sample the transmit signals from 1 and 3. The pseudoranges from 1 and 3 can then be differenced and adjusted by the difference in range between the two transmitters. This produces the clock offset between transmitters 1 and 3. By conducting this measurement more than once (from both transmitters 2 and 4), this error can be measured with greater precision. By repeating this process for every node on the network, the errors between every node and the true clock reference can be calculated.

The errors and propagation delay in a navigation system are outlined in the timeline shown in Figure 2.8. Here,

T_s = True time at which the signal left the transmit node

T_u = True time at which the signal reached the receive node

δt = Offset of transmit node from true time

t_u = Offset of the receive node from true time

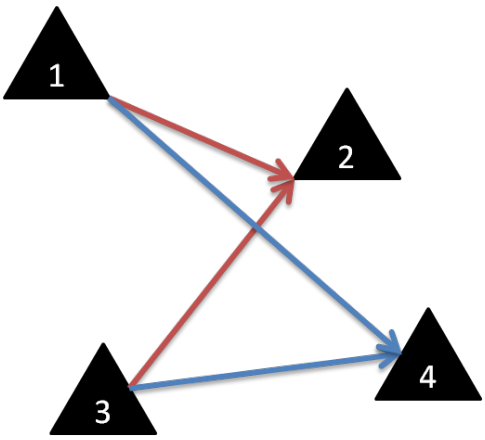


Figure 2.7: Sample of TDOA difference measurements to measure the transmitter clock errors.

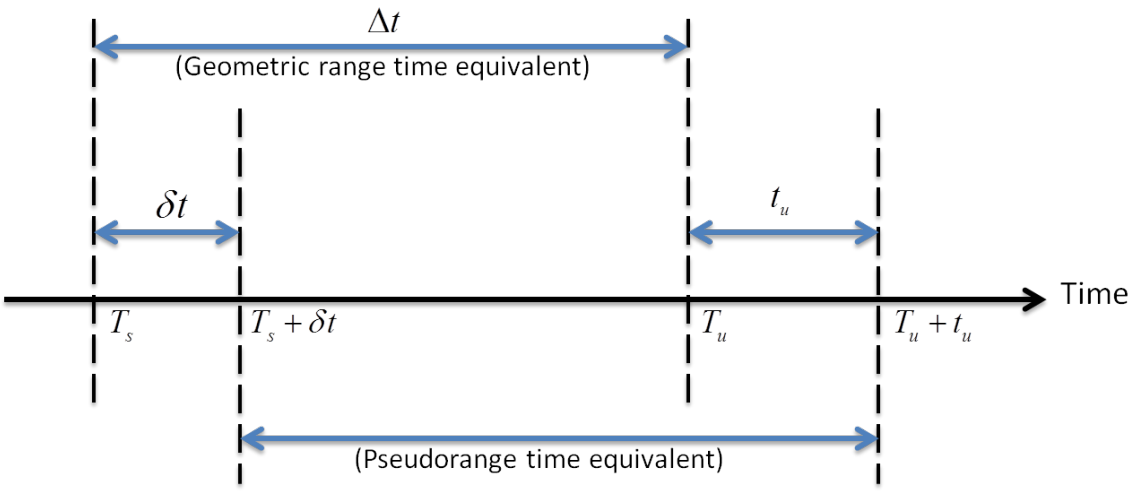


Figure 2.8: Diagram of the true timing and delays during a measurement with reference to the global timeline (reproduced from [6]).

The true geometric range and the psuedorange is thus

$$\text{Geometric range, } r = c(T_u - T_s) = c\Delta t \tag{2.13}$$

$$\text{Pseudorange, } \rho = r + c(t_u - \delta t) \tag{2.14}$$

$$\rho - c(t_u - \delta t) = ||\boldsymbol{s} - \boldsymbol{u}|| \tag{2.15}$$

where **s** is the transmitter location and **u** is the receiver location. Here, δt can be measured and accounted for utilizing TDOA techniques between the transmitters. By measuring δt, Equation (2.14) now simplifies to

$$\rho - ct_u = ||\boldsymbol{s} - \boldsymbol{u}||. \tag{2.16}$$

There are 4 missing parameters pertaining to the receiver: the 3-dimensional location (in Cartesian coordinates, (x_u, y_u, z_u)) and the receiver clock error, t_u. Each pseudorange is a function of these parameters and is given by

$$\rho_j = f(x_u, y_u, z_u, t_u) = \sqrt{(x_j - x_u)^2 + (y_j - y_u)^2 + (z_j - z_u)^2} + ct_u \tag{2.17}$$

where ρ_j is the pseudorange measurement from each transmitter, x_j, y_j, z_j is the location of that transmitter, and x_u, y_u, z_u is the user or receiver location. This produces a system of equations with the same unknowns: the receiver location and the receiver clock error. Note that this system is non-linear due to the square root, significantly increasing the difficulty in finding a solution.

The Least Squares Algorithm is utilized to solve this non-linear system of equations. This process involves the linearizion of the pseudorange equations followed by an iterative approach to solving for the receiver location and clock error. More advanced techniques, such as Kalman Filtering, can be implemented, but for the purpose of this thesis research the more simplistic approach will be taken.

For ease of derivation, the unknowns can be defined in vector notation as

$$\underline{\boldsymbol{x}} = \begin{bmatrix} x_u \\ y_u \\ z_u \\ ct_u \end{bmatrix}. \tag{2.18}$$

To accomplish the linearization of these equations, the Taylor series expansion will be utilized, given by

$$f(a + \Delta a) = f(a) + \Delta a \frac{df}{da} + \frac{(\Delta a)^2}{2!} \frac{d^2 f}{da^2} + \dots \tag{2.19}$$

For linearization purposes, higher orders above the first derivative will be neglected. One drawback to this method is that a point must be designated to linearize about. Thus, a guess at the receiver position and clock error must be injected into the system of equations and will be given as

$$\underline{\hat{\boldsymbol{x}}} = \begin{bmatrix} \hat{x}_u \\ \hat{y}_u \\ \hat{z}_u \\ c\hat{t}_u \end{bmatrix}. \tag{2.20}$$

An offset between the 4 unknowns of the receiver and the guesses at these values exists and can be written as

$$\left.\begin{aligned} x_u &= \hat{x}_u + \Delta x_u \\ y_u &= \hat{y}_u + \Delta y_u \\ z_u &= \hat{z}_u + \Delta z_u \\ ct_u &= c\hat{t}_u + \Delta ct_u \end{aligned}\right\} \quad \underline{\boldsymbol{x}}_u = \underline{\hat{\boldsymbol{x}}}_u + \Delta\underline{\boldsymbol{x}}_u. \tag{2.21}$$

Substituting these relationships into equation 2.17 produces

$$f(x_u, y_u, z_u, ct_u) = f(\hat{x}_u + \Delta x_u, \hat{y}_u + \Delta y_u, \hat{z}_u + \Delta z_u, c\hat{t}_u + \Delta ct_u). \tag{2.22}$$

Executing the first order approximation produces

$$
\begin{aligned}
&f(\hat{x}_u + \Delta x_u, \hat{y}_u + \Delta y_u, \hat{z}_u + \Delta z_u, c\hat{t}_u + \Delta ct_u) = \\
&\quad f(\hat{x}_u, \hat{y}_u, \hat{z}_u, c\hat{t}_u) + \frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, c\hat{t}_u)}{\partial \hat{x}_u}\Delta x_u + \frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, c\hat{t}_u)}{\partial \hat{y}_u}\Delta y_u \\
&\quad + \frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, c\hat{t}_u)}{\partial \hat{z}_u}\Delta z_u + \frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, c\hat{t}_u)}{\partial c\hat{t}_u}\Delta ct_u.
\end{aligned} \tag{2.23}
$$

Executing the partial derivatives produces

$$
\begin{aligned}
\frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, c\hat{t}_u)}{\partial \hat{x}_u} &= -\frac{x_j - \hat{x}_u}{\hat{r}_j} \qquad \frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, c\hat{t}_u)}{\partial \hat{y}_u} = -\frac{y_j - \hat{y}_u}{\hat{r}_j} \\
\frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, c\hat{t}_u)}{\partial \hat{z}_u} &= -\frac{z_j - \hat{z}_u}{\hat{r}_j} \qquad \frac{\partial f(\hat{x}_u, \hat{y}_u, \hat{z}_u, c\hat{t}_u)}{\partial c\hat{t}_u} = 1
\end{aligned} \tag{2.24}
$$

where,

$$\hat{r}_j = \sqrt{(x_j - \hat{x}_u)^2 + (y_j - \hat{y}_u)^2 + (z_j - \hat{z}_u)^2}.$$

Now the linearized range equation can be written as

$$\rho_j = \hat{\rho}_j - \frac{x_j - \hat{x}_u}{\hat{r}_j}\Delta x_u - \frac{y_j - \hat{y}_u}{\hat{r}_j}\Delta y_u - \frac{z_j - \hat{z}_u}{\hat{r}_j}\Delta z_u + \Delta ct_u. \tag{2.25}$$

Simplification produces a more concise form

$$\Delta\rho_j = a_{xj}\Delta x_u + a_{yj}\Delta y_u + a_{zj}\Delta z_u - \Delta ct_u \tag{2.26}$$

with

$$
\begin{gathered}
\Delta\rho_j = \hat{\rho}_j - \rho_j \\
a_{xj} = \frac{x_j - \hat{x}_u}{\hat{r}_j}, \quad a_{yj} = \frac{y_j - \hat{y}_u}{\hat{r}_j}, \quad a_{zj} = \frac{z_j - \hat{z}_u}{\hat{r}_j}.
\end{gathered}
$$

Now the linearized equations for the range error over the n different pseudorange measurements to each transmitter is

$$\begin{aligned}
\Delta\rho_1 &= a_{x1}\Delta x_u + a_{y1}\Delta y_u + a_{z1}\Delta z_u - \Delta ct_u \\
\Delta\rho_2 &= a_{x2}\Delta x_u + a_{y2}\Delta y_u + a_{z2}\Delta z_u - \Delta ct_u \\
\Delta\rho_3 &= a_{x3}\Delta x_u + a_{y3}\Delta y_u + a_{z3}\Delta z_u - \Delta ct_u \\
&\vdots \\
\Delta\rho_n &= a_{xn}\Delta x_u + a_{yn}\Delta y_u + a_{zn}\Delta z_u - \Delta ct_u.
\end{aligned} \tag{2.27}$$

Converting to matrix form for display and computational ease produces

$$\Delta\underline{\boldsymbol{\rho}} = \underline{\underline{\boldsymbol{H}}}\Delta\underline{\boldsymbol{x}}$$

$$\Delta\underline{\boldsymbol{\rho}} = \begin{bmatrix} \Delta\rho_1 \\ \Delta\rho_2 \\ \Delta\rho_3 \\ \vdots \\ \Delta\rho_n \end{bmatrix} \qquad \underline{\underline{\boldsymbol{H}}} = \begin{bmatrix} a_{x1} & a_{y1} & a_{z1} & -1 \\ a_{x2} & a_{y2} & a_{z2} & -1 \\ a_{x3} & a_{y3} & a_{z3} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{xn} & a_{yn} & a_{zn} & -1 \end{bmatrix} \qquad \Delta\underline{\boldsymbol{x}} = \begin{bmatrix} \Delta x_u \\ \Delta y_u \\ \Delta z_u \\ \Delta ct_u \end{bmatrix} \tag{2.28}$$

For the Least Squares method, there are three general cases which can occur:

- $n < 4$: Is the under-determined case

  Cannot solve for $\Delta\underline{\boldsymbol{x}}$

- $n = 4$: Uniquely determined case

  Generally one valid solution for $\Delta\underline{\boldsymbol{x}}$ solved by calculating $\underline{\underline{\boldsymbol{H}}}^{-1}$ in $\Delta\underline{\boldsymbol{x}} = \underline{\underline{\boldsymbol{H}}}^{-1}\Delta\underline{\boldsymbol{\rho}}$

- $n > 4$: Overdetermined case

  No perfect solution, solve iteratively (Least Mean Squares adaptive algorithm)

In the case that there are 4 transmitters utilized to form a particular measurement, a single solution is available. This solution is found by solving for $\Delta\underline{\boldsymbol{x}}$ then utilizing

Equation (2.21) to find the location and clock error of the receiver. For the case having more than 4 transmitters, the solution requires an iterative approach as visualized in Figure 2.9. This iteration is conducted until the distance step, $\Delta\underline{\boldsymbol{x}}$ falls under a set stepping threshold meaning the solution has converged onto a final set of values for the location and clock error [6].

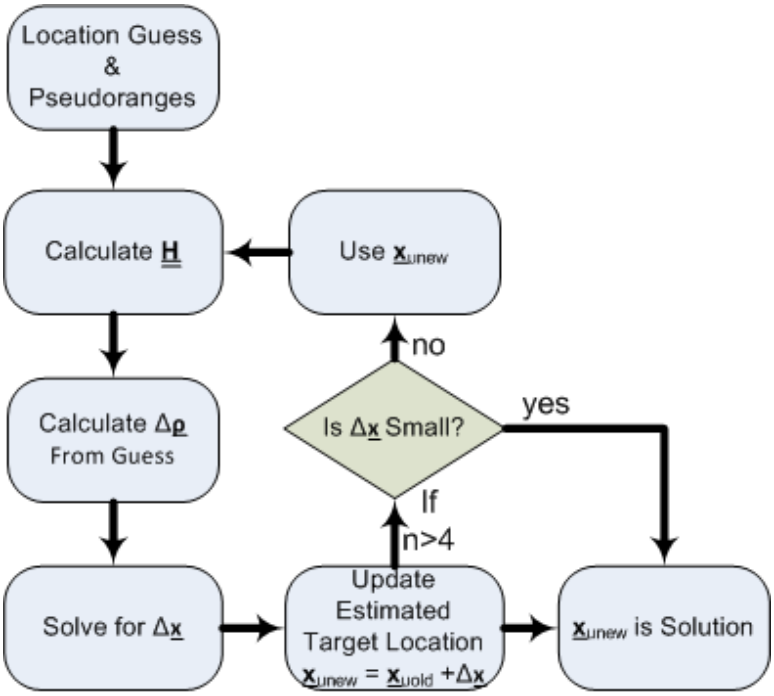


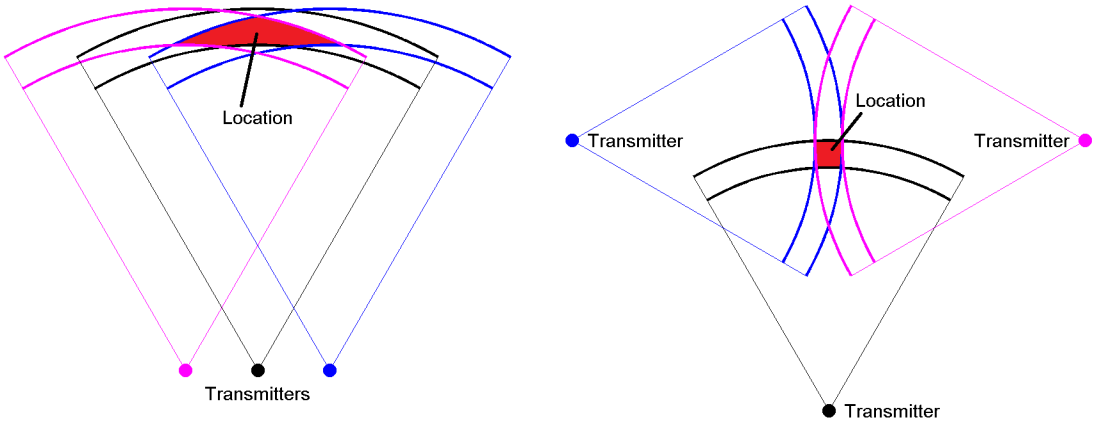
Figure 2.9: Flow Diagram of the process by which a solution is obtained utilizing an iterative approach from the Least Squares Algorithm.

2.3.2 System Navigation Accuracy.

It is desired to know the potential navigation accuracy of the NoNET when utilizing the Least Squares algorithm approach to solving for the receiver position. Assuming NoNET captures perfect pseudorange measurements and clock error corrections, the resolution of each one-way range measurement is 0.2 meters or approximately 0.66 feet. This range resolution is driven by the sampling rate of the NTR ADCs. The sampling rate

utilized in this case is 1.5G samples per second. If perfect pseudorange measurements with NTR are only accurate to 0.2 meters, how much of an impact does this make on the final receiver position solution?

The final accuracy is highly dependent upon the measurement geometry utilized. Figure 2.10 shows an example of how a poor geometry can significantly effect the final results. Error in one pseudorange measurement impacts the solution on a greater scale if the measurement geometry is less than optimal.



(a) Poor geometry

(b) Better geometry

Figure 2.10: Examples of the impact system geometry can have on the final navigation measurement results.

One can view any errors in the pseudorange measurements as errors residing within a measurement domain. It is necessary to know the impact these measurement domain errors have in the final position domain. In the Least Squares algorithm, we need to know the

co-variance matrix

$$\boldsymbol{C}_x = \begin{bmatrix} \sigma^2_{x_u} & \sigma_{x_u,y_u} & \sigma_{x_u,z_u} & \sigma_{x_u,\delta t_u} \\ \sigma_{x_u,y_u} & \sigma^2_{y_u} & \sigma_{y_u,z_u} & \sigma_{y_u,\delta t_u} \\ \sigma_{x_u,z_u} & \sigma_{y_u,z_u} & \sigma^2_{z_u} & \sigma_{z_u,\delta t_u} \\ \sigma_{x_u,\delta t_u} & \sigma_{y_u,\delta t_u} & \sigma_{z_u,\delta t_u} & \sigma^2_{\delta t_u} \end{bmatrix}, \tag{2.29}$$

where the diagonal terms are the variances of the particular receiver locations in all three dimensions as well as the variance in its clock error. The off-diagonal terms are the co-variances between each of these four parameters providing a way to quantify cross-dependences between each of the result dimensions.

To obtain the variances in the receiver unknowns, the co-variance matrix of the pseudorange measurements is used, given by

$$\boldsymbol{C}_\rho = \begin{bmatrix} \sigma^2_{\rho_1} & \sigma_{\rho_1,\rho_2} & \cdots & \sigma_{\rho_1,\rho_n} \\ \sigma_{\rho_1,\rho_2} & \sigma^2_{\rho_2} & \cdots & \sigma_{\rho_2,\rho_n} \\ \vdots & \vdots & \ddots & \sigma_{\rho_3,\rho_n} \\ \sigma_{\rho_1,\rho_n} & \sigma_{\rho_2,\rho_n} & \sigma_{\rho_3,\rho_n} & \sigma^2_{\rho_n} \end{bmatrix}. \tag{2.30}$$

From Least-Squares,

$$\boldsymbol{C}_x = \left(\boldsymbol{H}^T \boldsymbol{C}_\rho^{-1} \boldsymbol{H}\right)^{-1}. \tag{2.31}$$

In order to determine the final variances in the navigation solution, or the error in a rough sense, it's necessary to know the geometry and variances present in the pseudoranges due to multipath, sampling rate limitations, and other phenomena. It is extremely difficult to determine a fixed number for system accuracy, as it is dependent on many different factors and can change for every measurement geometry.

2.4 Summary

In this Chapter, the fundamental theories and hardware configurations which form the foundation for navigation with NTR were described. These principles can be found in

all work conducted for this research. The next chapter will outline how the research was conducted and each of the experimental configurations.

III. Methodology and Configuration

This chapter outlines the process and experimental configurations by which this research was conducted and analyzed. Research began with a focus on the development of the navigation algorithm for use in the final NoNET software package and its simulations. The algorithm was followed by software modifications to the NoNET. During this time, hardware upgrades to the system were underway, which prevented immediate testing of these software modifications. Thus, attention was shifted to the remote triggering aspect of this research, whereby different methods to issue the global synchronized trigger to every node on the network were considered and tested. Finally, with upgraded hardware, debugging of the navigation software package was accomplished, followed by multiple navigation tests and an indoor multipath impact analysis.

3.1 Navigation Algorithm

Research began by analyzing various navigation and localization techniques. Coding and simulation of the navigation algorithm in MATLAB commenced. Inputs to the algorithm include the pseudoranges and the initial guess for linearization purposes. Outputs from the algorithm are the final estimations of the receiver or user location and the receiver's clock error. A full simulation tool to verify correct functionality of this developed algorithm was written in MATLAB. This simulation tool simulated all NTR waveforms, as well as, the propagation time delays to form synthetic pseudoranges. Figure 3.1 shows the user interface for this tool.

The simulation tool was useful for experimentation with the various placement geometries of the networked transmitters. It also showed the impacts of various levels of pseudorange measurement errors on the final location accuracies.

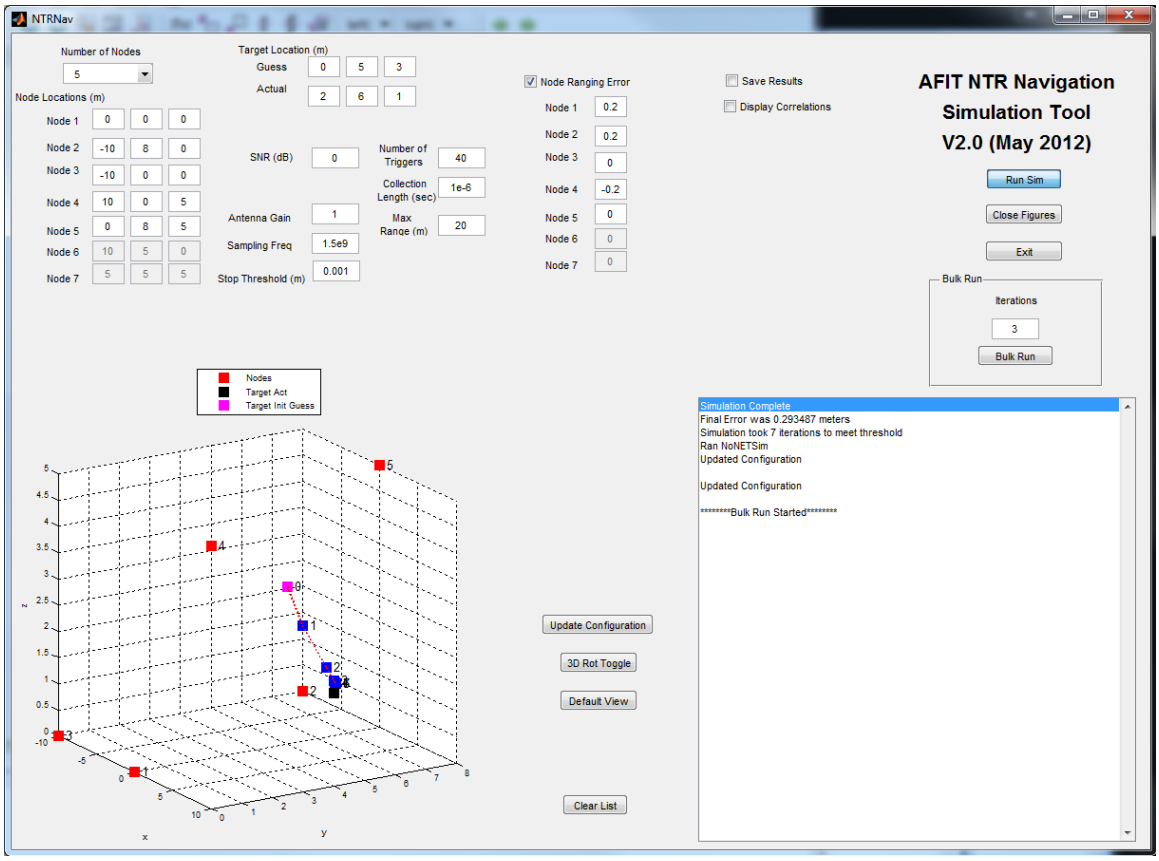


Figure 3.1: Screenshot of the constructed navigation algorithm simulation tool.

3.2 NoNET Software Package

Because the NoNET hardware is primarily software configurable, software is the main focus of this research. Transitioning from a two-way time delay measurement system to a one-way time delay system utilizes the same pre-existing capture process in software. The difference between the two-way and one-way software lies in how the capture triggering is handled, and how the data is handled after capture. During two-way time delay measurements, global triggers need not be perfectly synchronized as each NoNET node utilizes its own transmission for correlation. Thus, each node on the network can configure its hardware and issue its own software-based trigger on its own time-line. During navigation capture or a multistatic/bistatic radar scenario, two things must

be accomplished. The nodes must respond to a synchronized trigger and each NoNET node requires the transmit data from the other nodes for correlation processing. Thus, the waveform data is distributed to all other nodes on the network after the global synchronized capture takes place.

Further details of the triggering portion of this system are provided in Section 3.3.

3.2.1 Remote MATLAB Control.

One requirement for navigation was the development of a method by which all computers on the NoNET could be controlled remotely and share large amounts of data quickly. Previous research utilized a file-watching service in MATLAB to control computers on the network. Data-sharing was accomplished by writing files to remote hard drives in Windows. This method was cumbersome, because it required saving the capture data and then loading the data into the MATLAB workspace. With a single capture from one node reaching upwards of 50MB in size, this process is slow at best. A new method was required to obtain a reasonable refresh rate during the navigation captures.

Because preexisting hardware and software interfaces for the NoNET were accomplished in MATLAB, it was desired that the solution to the remote management problem be accomplished in MATLAB as well. No fluent experience in any other programming language or even integration of other languages into MATLAB was readily available. A potential solution to this quandary was MATLAB's Parallel Computing Toolbox. This toolbox allows a user to manage remote processing jobs and MATLAB workers from a single remote terminal. Two drawbacks to this toolbox prevented its use in the final NoNET navigation software. The first was the cumbersome methods by which these remote workers were setup. The second drawback was that there was little control over which remote computer a particular MATLAB worker was operating. Due to this lack of capability, the Parallel Computing Toolbox for MATLAB was eliminated as a useful solution to the remote control dilemma.

The *tcpip* method was eventually developed to provide remote control over MATLAB. In summary, the *tcpip* function provides a serial link between two instances of MATLAB, allowing for direct distribution of data from one MATLAB workspace to another. This link was utilized for control purposes, on top of data transfer, by passing serial character strings to the remote nodes. These remote units read the string and compare it to a list of pre-determined commands. Based upon the received string, the remote MATLAB instances execute scripts for that particular command. For virtually unlimited control over the remote nodes, each node is programmed so that if no string compare match is found in the list of commands, an attempt to execute an *eval()* command of the received string is made. Thus, the controlling MATLAB instance can send virtually any execution command to all remote nodes on the network of which *tcpip* links have been established. This control method is extensively detailed in Appendix A of this document.

3.2.2 The Global Capture Sequence.

After control over remote instances of MATLAB was accomplished, the capture sequence and data processing portions of the software required attention. With the goal of utilizing the preexisting NoNET hardware, a fixed sequence of actions executed simultaneously on every active node across the network is required in order to perform a networked capture. Figure 3.2 provides a visual representation of this process.

Each node on the network must step through the capture card configuration process and arming before the global trigger is issued. Verification that each remote capture card has successfully armed prior to issuing the global trigger aids in the mitigation of system errors, where the trigger is issued before a capture card is fully armed.

After issuing the global trigger, the controlling node on the network manages the data distribution process. This process steps through each node on the network and distributes its transmit data to all networked nodes. After data distribution, the network transitions to the processing stage, where each node calculates pseudoranges via correlation against

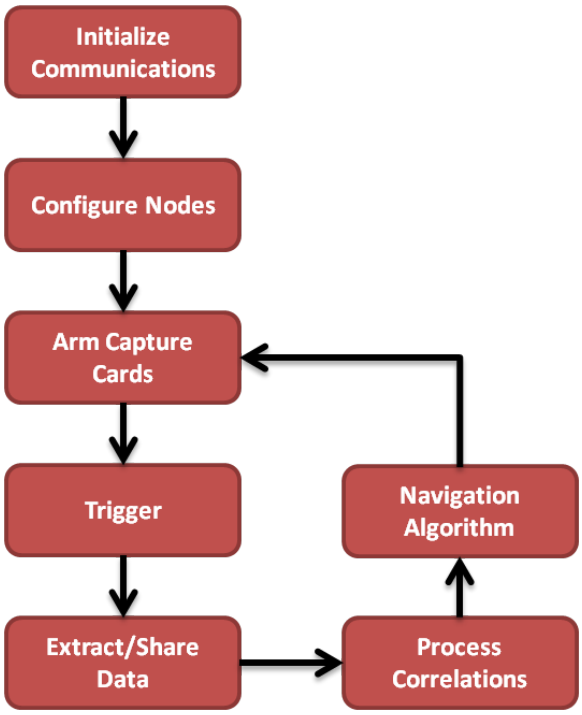


Figure 3.2: Diagram of the global sequence of events required to perform a networked data capture and receiver localization across all NoNET nodes.

every other node's transmit data that was shared in the last step. If the node is functioning as a transmitter, it focuses these calculations toward determining the trigger timing errors between each of the other transmitters. These errors are then passed to the receiver for later correction of its measured pseudoranges. Finally, the receiver processes correlations to obtain its pseudoranges. It then uses the timing errors calculated by the transmitters to produce final estimations of its three dimensional location and trigger/clock error. Note that this entire process is controlled and managed via a central control computer, which in the case of this experimentation, also served as the receive node.

3.3 Triggering Synchronization

The ADC capture trigger must be synchronized for one-way measurements across the network such that the transmission from a remote node on the network is captured

synchronously by that node's ADC and the ADC on the receiving node. A visual example of this trigger synchronization requirement is provided in Figure 3.3. Without overlap in the transmit ADC's capture and the receive ADC's capture, each set of data would provide no useful information to the user after correlation.

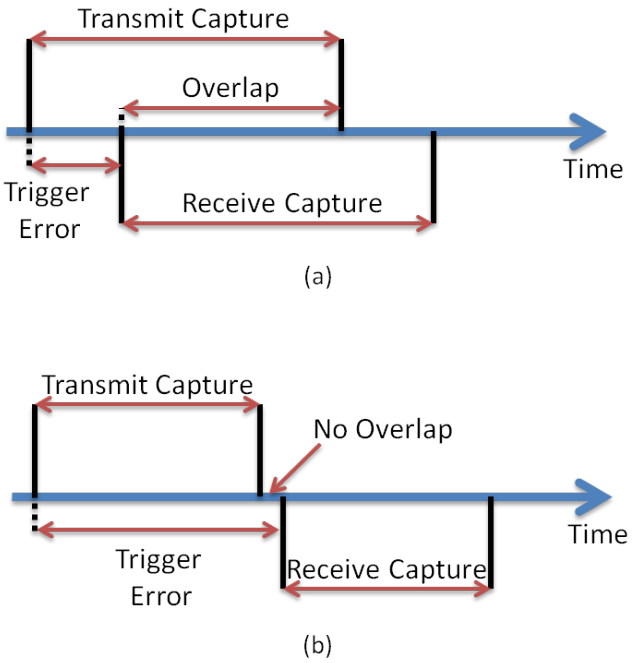


Figure 3.3: Global capture timeline (a) showing an acceptable capture where the ADC triggers were adequately synchronized and (b) showing an unacceptable capture where the ADC triggers were not synchronized and data cannot be correlated.

A couple methods by which the current ADCs can be triggered was explored for use in the navigation software with varying degrees of success and practicality. With the current set of ADC card drivers, only 14ms of data can be captured at one time. This limitation sourced a requirement to produce a wireless method which provides a synchronized trigger pulse having less than 7ms of timing error.

3.3.1 Wired.

The wired approach to issuing a trigger to the ADCs is undoubtedly the simplest, but arguably the least practical for final implementation. Previous projects with NoNET utilized a coaxial connection via the external BNC connector on the ADC supplied by an external function generator to produce the electronic trigger pulse. This method works well for short node separation distances, but becomes much less practical when greater separation distances with a greater number of nodes is introduced. Several hundred feet of coaxial cable not only becomes unwieldy, but it can be exceedingly expensive.

An alternative to this trigger method is to remain with the hardwire approach utilizing an external function generator, but substitute a less cumbersome, less expensive cable than the coaxial. The ADCs contain an on-board header which accepts a TTL 5V trigger pulse as an alternative to the BNC connection. Connecting to this header requires a simple two-conductor cable: one for ground, another for the signal. This two-conductor cable was purchased in a large spool of 1000ft for much less than coaxial cable. Each node was then supplied with approximately 100ft of trigger cable length which is more than adequate for any testing distances during this thesis research. The function generator was linked to the controlling computer via RS-232 and controlled via serial commands within MATLAB. Figure 3.4 diagrams the final trigger connections. While wireless trigger methods were researched, this hardwire method was utilized in the final navigation attempts with the NoNET due to its simplicity and ease of debugging. This method does, however still maintain a sense of impracticality.

3.3.2 Wireless.

The wireless approach to providing a synchronized trigger requires a more complex solution, but is more practical for use in a final system than running cable between each node. There are two degrees of wireless implementation that were researched: (1) computer

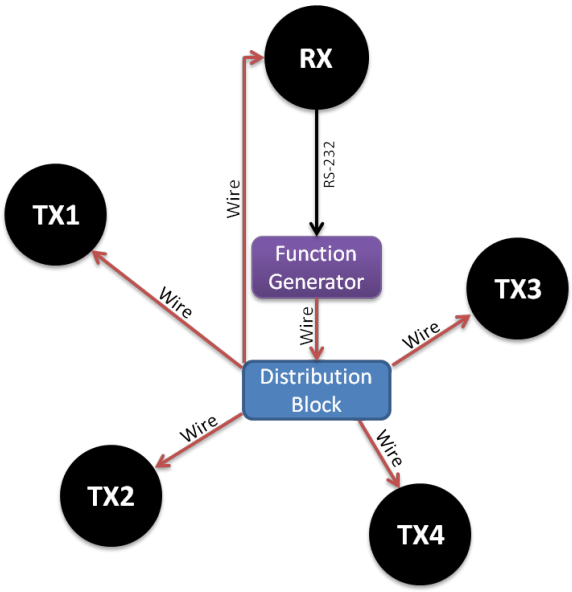


Figure 3.4: Block diagram of the implemented triggering scheme. Each wire consists of two-conductor cable connected to the internal SMC header on-board each ADC card.

controlled triggering with no additional hardware and (2) a separate wireless module connected to the triggering ports of the ADC cards.

The easiest implementation of wireless triggering is the issuance of a software trigger to the ADCs at a designated and synchronized computer clock time. The question that remains is whether or not the computer clocks across the network can be synchronized to the required trigger accuracies. These computer clock accuracies must also be maintained at their levels indefinitely. In order to test this method, two NoNET computers were networked and setup with Windows Network time protocol (NTP) service. The offset between these clocks was tracked overnight. Results of this test can be found in Chapter 4.

The second method for implementing a wireless trigger utilizes a separate external device. This device provides an output pulse to the ADC cards, but synchronizes the issued pulse wirelessly. The results of research conducted on this second method produced a micro-controller (MCU)-based clock system which communicates wirelessly

using Xbee wireless communication modules. These devices were structured as a master-slave network. This means that one unit of the developed hardware serves as a master clock of which all other slave units synchronize to (Figure 3.5).

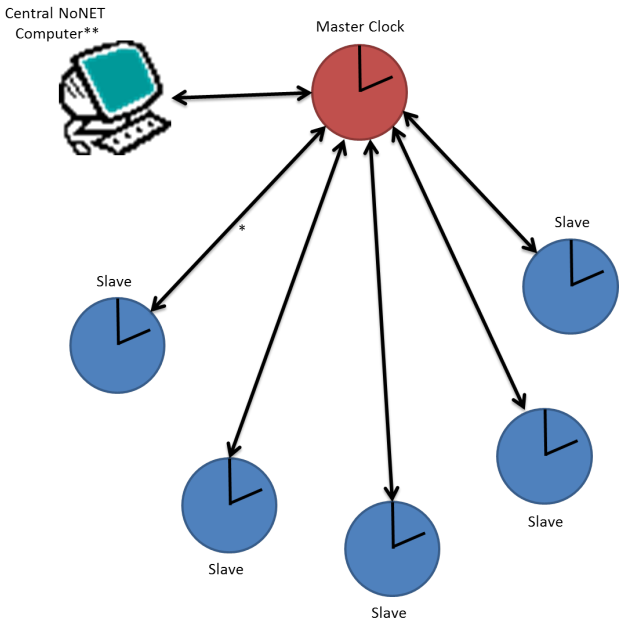


Figure 3.5: Structure diagram showing the master-slave relationship of the NoNET wireless clocks.

This clock network utilizes the synchronization process outlined in Figure 3.6. This synchronization process begins with each master transmitting a time hack signal to each slave at the rising edge of the output pulse. The rising pulse edge is essentially time zero. The slave receives this time hack with an accrued network delay from the Xbee devices. To estimate this network delay, the slave clock then pings the master unit and measures the response time. It was assumed that the time it takes the Xbee to process and transmit a signal from the slave to the master is identical to the time it takes to process and transmit a signal from the master to the slave. Thus, the round trip network delay could be divided in half in order to estimate the network delay accrued on top of the time hack. It was

determined that this synchronization had to occur approximately every 8 minutes to account for drift.

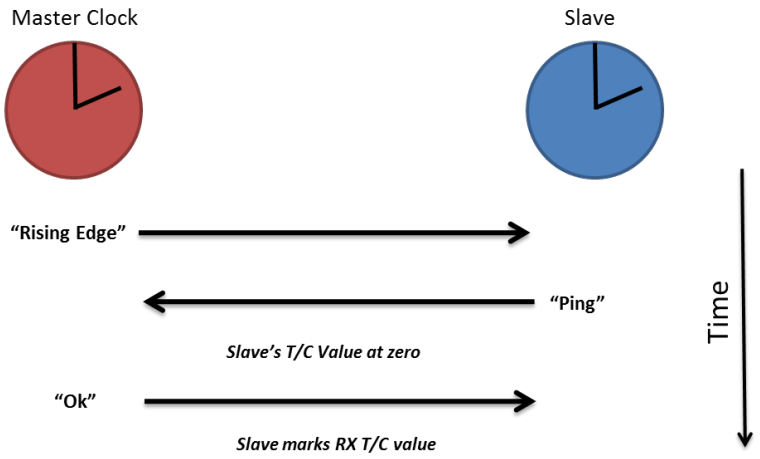


Figure 3.6: Diagram of the implemented synchronization procedure utilized on the NoNET wireless clocks.

The NoNET wireless clocks were constructed utilizing an ATMEGA168 MCU clocked with a 10PPM high-accuracy crystal. The MCU hardware timer/counter registers were utilized for their stability because they operate independently of the software that is being executed in its CPU. The final clock design required logic conversion between the 5V MCU and the 3.3V Xbee module as well as internal voltage regulation. The ability to utilize either an external power adapter or an internal battery was added. Figure C.1 shows the hardware schematic of these wireless clocks. The constructed units are depicted in Figure 3.7.

Final measurements on the accuracies and analysis of any errors were accomplished utilizing a combination of a digital oscilloscope and a logic analyzer. The oscilloscope was connected to the output pins of three devices (one master and two slaves) and was utilized to track the slave pulse offsets from the master's over time. Triggering on the oscilloscope was

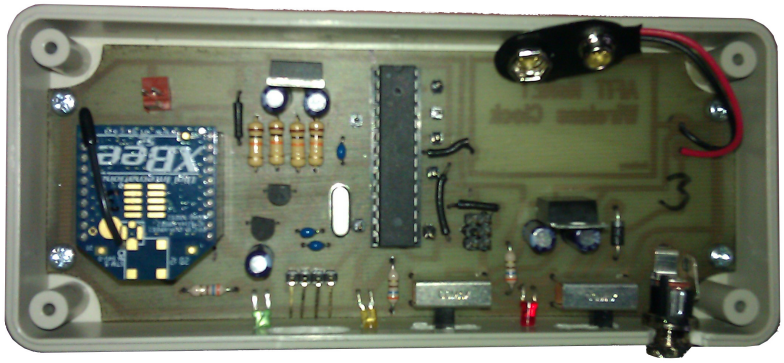


Figure 3.7: Picture of the finished NoNET wireless clocks.

set to the rising edge of the master's output pulse. Measurements of the Xbee network lags were taken utilizing the logic analyzer sampling two devices (one master and one slave). Four connections from the logic analyzer were made to the clocks: two to the master and two to the slave. The two connections on each clock were the Xbee transmit and receive data pins. This configuration allowed for measurement of the time between when a MCU sends data to be transmitted and when that specific data is output from the receive pin at the other end of the Xbee link.

3.4 Indoor Multipath Experimentation

A major complexity in performing navigation via simple correlation of the transmit and receive waveforms to produce range measurements is the mitigation of multipath. In cluttered environments, a receiver not only receives the direct path transmission, but reflections of this signal off of a multitude of objects and walls in an indoor environment. In some cases, locating the proper correlation peak for use out of the many returns can

become difficult. In order to quantify the effects this multipath has on the final system performance, experimentation to capture these effects was performed.

For these measurements, a one-way capture was accomplished utilizing a single NTR unit. One antenna was placed on one side of a plain indoor brick wall. This antenna served as the transmitter. Within the room on the other side of the wall, the second antenna of the NTR unit was placed to act as a receiver. The range error produced by the extra length of cable required to perform this measurement was calibrated out of the final results by connecting the two antenna leads together and performing a capture. The only portion of the system not calibrated out utilizing this method were the antennas. Figure 3.8 depicts the measurement setup.

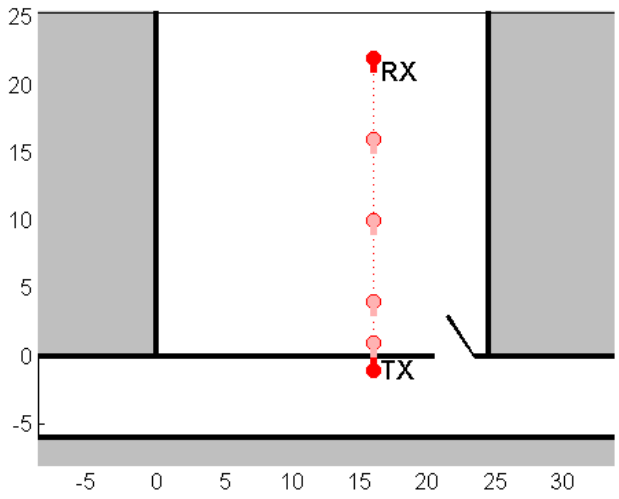


Figure 3.8: Depiction of the configuration utilized to quantify the level of multipath present when transmitting in an indoor environment through a standard 6” thick brick wall.

3.5 Indoor One-way Range Experimentation

An important component to indoor navigation utilizing NoNET is the capability to perform accurate range measurements with one-way signal transmissions. The process

under test involves a noise transmission between the nodes, transferring the transmitted waveform data to the receiving node, and performing the correlations between the data sets. In order to conduct this experiment, two nodes were setup at a fixed distance and triggered to capture simultaneously.

This capture was calibrated by connecting the transmit antenna lead on one node to the receive antenna lead on the second node. This calibration capture accounts for the timing lag produced by all internal components and external antenna cabling between the transmit and receive paths of the two units. Again, the only portion of the system not calibrated for were the antennas.

Having conducted this one-way range capture in an indoor environment, the full effects of multipath were also part of the final results for this experiment. This measurement produced a realistic representation of the correlation, which was to be encountered during the navigation phase of this research before setting up the full tracking experiments.

3.6 Indoor Navigation Capture Experimentation

After each subsection of the NoNET navigation software package was debugged to the greatest extent possible, attempts to setup all available nodes and perform localization of the receive node were conducted. The locations for these experiments included the AFIT Advanced Navigation Technology (ANT) Center's laboratory area and Kenney Hall Auditorium. The ANT Center's laboratory area contained an open area which allowed for separation of the transmitter nodes. Each transmitter was placed along the outside edges of the testing area. The receiver/central node was then placed at varying positions within the room. An example of one of the different tested geometries is provided in Figure 3.9 as well as a picture of the setup nodes in Figure 3.10.

The testing conducted in Kenney Hall Auditorium was aimed at measuring the multipath effects in an open indoor environment with a lower level of clutter. The briefing

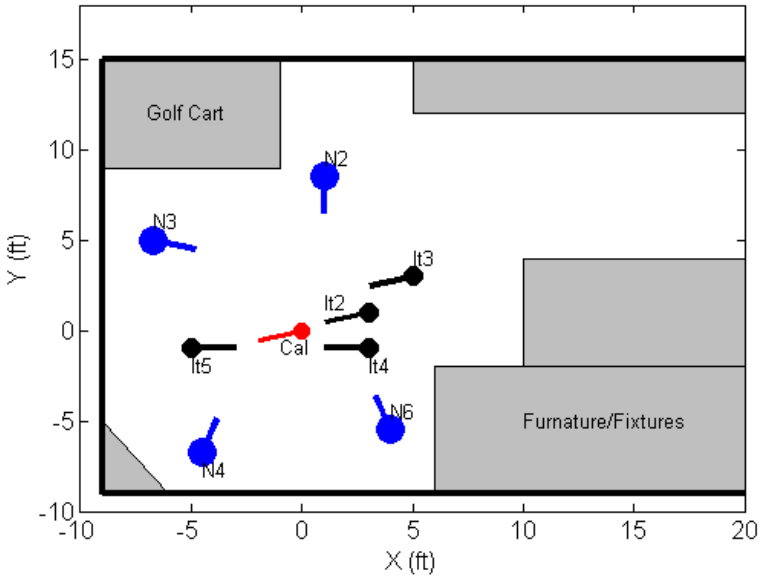


Figure 3.9: Depiction of the configuration utilized during the indoor navigation experimentation in the AFIT ANT Center's lab space showing the placement (dots) and orientation (lines) of each node.

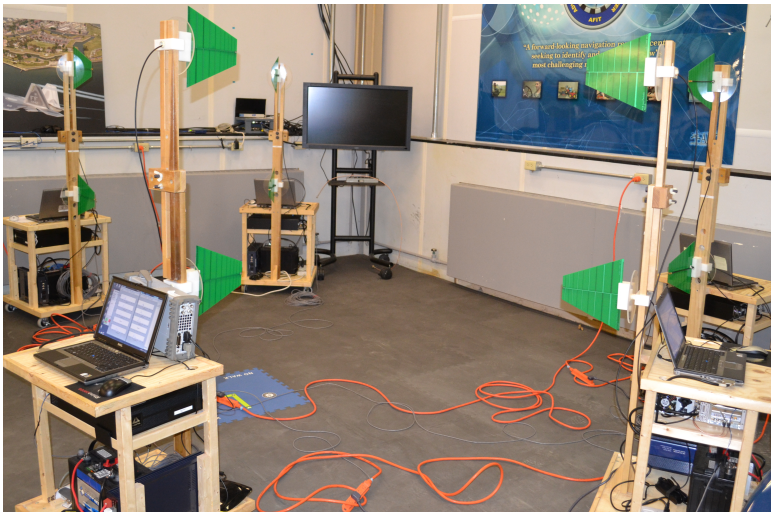


Figure 3.10: Picture of the ANT Center indoor navigation testing.

stage of this auditorium provided for just such an open environment. Figure 3.11 shows a picture of this testing configuration.

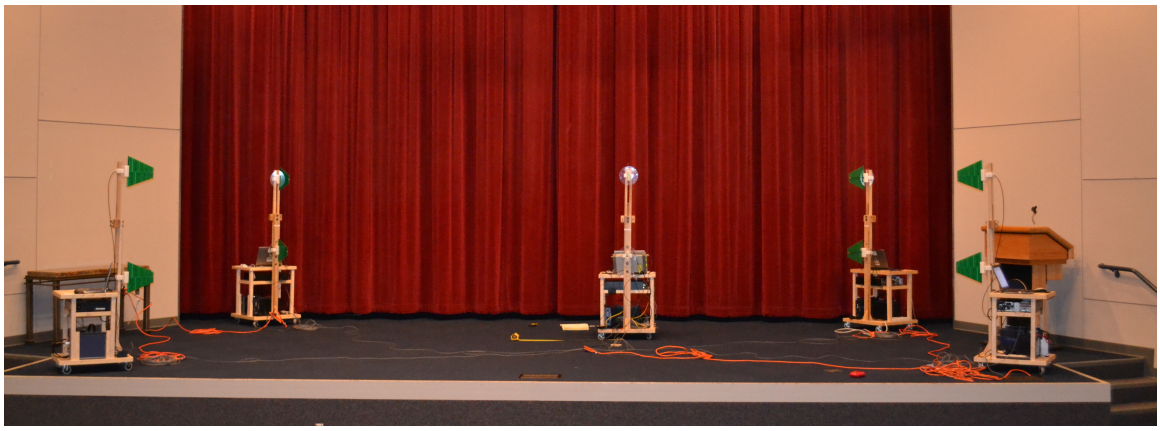


Figure 3.11: Picture of the Kenney Hall indoor navigation testing.

Due to the complexity of setting up transmitter antennas at elevated heights, each of the measurements were performed in the same elevation plane. Thus, all measurements conducted were two dimensional but still serve to demonstrate system performance in indoor environments.

Multiple global captures took place at each receiver location. The number of triggers per capture, the length of each capture, the antenna polarizations, and the transmit power attenuation levels were among the various capture parameters that were modified. By varying these capture parameters, links to system performance based on different parameter selections were established. Passive captures to measure the background electronic environment were also taken to analyze signal to noise levels during testing.

3.7 Outdoor Navigation Capture Experimentation

The last experiment conducted for this thesis research involved operation in an outdoor environment. The purpose behind conducting navigation tests in this environment was to

form comparisons between it and the indoor tests. The correlations produced in the indoor environments contained many returns from the vast level of multipath producing error in the final results. In an outdoor environment, correlations consist solely of the ground-bounce multipath with no wall or other clutter reflections. Here the two correlation results could be compared producing a measure as to the level at which multipath affects the final NoNET navigation performance.

A better sample of the RF environment in this frequency range was desired as well. This was due to the relatively large amount of radio frequency interference (RFI) seen during the indoor testing. Outdoor testing, however, received higher levels of this interference which degraded system performance.

Configurations for this test remained similar to that of the indoor test. Transmitters were spread out over an open area pointing toward a central location where the receiver was positioned. Various captures were conducted with the receiver at different locations.

3.8 Summary

In this Chapter, the methodology for experimentation was described. The desired data from each experiment was also described. The results and analysis from each experiment are provided in Chapter 4.

IV. Results and Analysis

This chapter provides the documentation and final analysis of the results obtained during this research. The results from each experiment described in Chapter 3 will be outlined, as well as the detailed analysis of any unexpected errors or phenomena. The attempts to improve system performance will also be described.

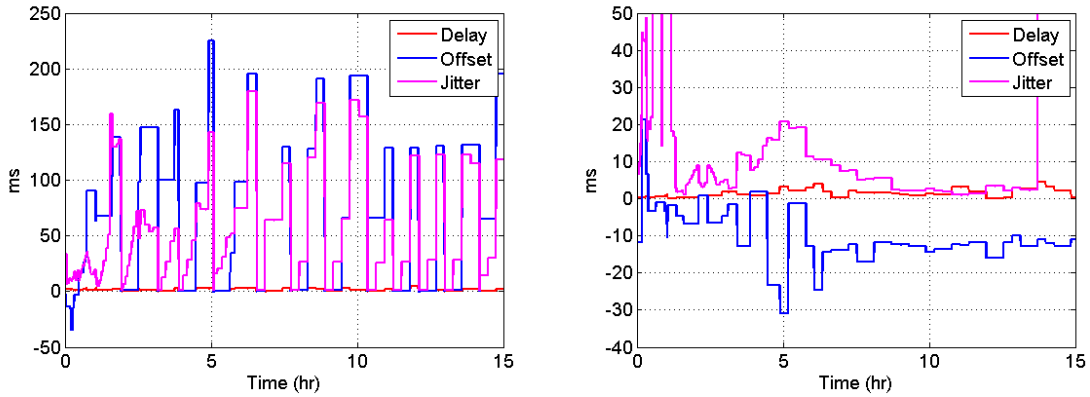
4.1 Trigger Synchronization

The results in this section include the initial research concering the global capture synchronization between each of the NoNET nodes. First, the results from testing the controlling computer clocks as the trigger sources will be discussed. Then the final performance specifications on the NoNET wireless clock units will be analyzed. Remember that the design threshold for any solution to this synchronization dilemma must have less than 7ms of synchronization error. This requirement was sourced from the fact that under the current configuration, the NoNET ADC capture cards are not capable of capturing and uploading more than 14ms at a time to the connected computer.

4.1.1 Software Triggering.

As described in Section 3.3, experimentation with Microsoft Window's computer clock synchronizations was conducted. The hypothesis under test was whether or not a standard Window's computer clock could be synchronized to another Window's computer clock and maintained at an offset of less than the design threshold of 7ms. By configuring the Windows' NTP service on two NoNET laptops networked over an ad-hoc wireless connection and monitoring the service results in MATLAB, results could be tracked over the course of approximately a 15 hour period (Figure 4.1).

Two tests were monitored, the first utilized the NTP service as intended. In other words, the service was activated and left untouched (Figure 4.1a). The second test



(a) Uncorrected

(b) Periodic Correction

Figure 4.1: Results of computer clock synchronization attempts utilizing Windows' NTP Service

attempted to improve these results. It was noted that the service begins by synchronizing at quick intervals, but slows its refresh rate over time, producing large clock offsets in between corrections. It was hypothesized that by resetting the NTP service at regular internals, thereby forcing it to continuously update, the final offset results between the two computer clocks could be improved.

This hypothesis was proved correct. By resetting the service at regular intervals, offset was reduced by an order of magnitude. The results pictured in Figure 4.1b show that over time, the offsets utilizing this periodic correction will settle to approximately 10ms. This, however, would take several hours of setup if implemented, not to mention the fact that even with this success, 10ms does not meet the 7ms threshold aforementioned.

A quick search of various internet forums and other resources showed techniques claiming to reach sub-millisecond clock synchronization accuracies. However, these techniques mostly utilized a GPS receiver linked to their computers which provided a much more accurate time base. Since the main purpose of this project was to provide

an alternative to the GPS, it is not feasible to link the controlling computers to the GPS, not to mention most testing must be accomplished in indoor environments, where GPS is not available.

Another question encountered was exactly how software triggering would be accomplished with the synchronized computer clocks. MATLAB might not have the capability to execute a command at an exact moment in time. Specifically, this execution timing would need accuracy down to the millisecond. While operating NoNET on a Windows operating system, would Windows also impede this command accuracy? There were too many questions to be answered in order to utilize this method of network triggering, not to mention the failed computer clock synchronization attempts.

4.1.2 Wireless Triggering.

The work on wireless implementation of a global trigger was one of the main portions of the work conducted for this research project. As described in Chapter 3, the NoNET wireless clock modules were designed and constructed, then final quantification and analysis of any error between the nodes was measured and conducted.

The first measurements were those tracking the error between the clock outputs over time with the oscilloscope. This determined the final accuracy of these wireless devices. Figure 4.2 shows the output measurements between the master and two of its slaves after approximately 3 hours of running. Tracking of the delay seen here between the two slaves and the master never exceeded approximately 4ms. Between each clock adjustment, the offsets between the wireless slaves and the master gradually increases due to drift in each unit.

The drift between the units occurred over time at a rate of approximately 75 microseconds per minute. With this drift rate, synchronization between the MCUs is accomplished every 8 minutes. This maintains the 4ms accuracy of the system.

Two questions remained at this point in the wireless clock analysis:

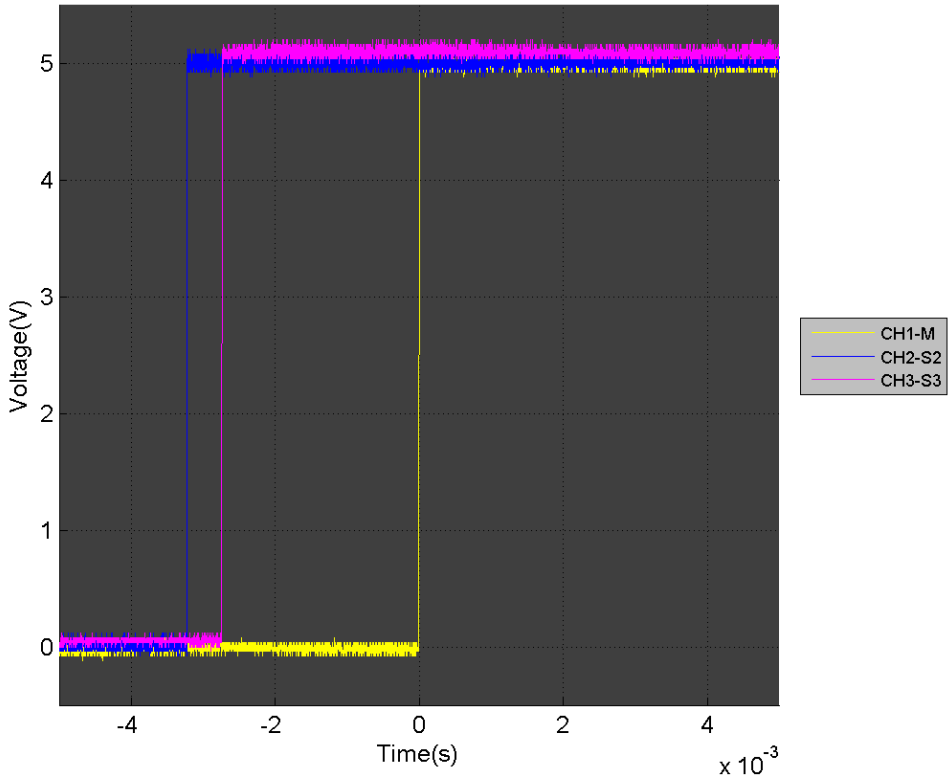


Figure 4.2: Oscilloscope measurement of the offset between the wireless clocks after 3 hours of runtime utilizing MCU software v4

- Why is drift occurring when high accuracy crystals and hardware counters are being utilized?

- What is causing the final synchronization errors between the devices and why is it not consistent over time?

To address the first question, further research into high accuracy clocks and oscillators was conducted. It was determined that the 10ppm crystals implemented onto these devices may be the root cause. These devices may not be performing as anticipated, due to not having a temperature controller built in. Accuracy and stability of these crystals is highly dependent upon them maintaining a state of thermal equilibrium. Also, slight variations

between the MCUs may be causing some additional drift. Variance in the capacitance utilized to resonate the clocking crystals could be causing slight offsets, as well as variances in the time it takes to address the hardware timer/counter interrupts. A potential solution to this issue in the future would be to eliminate the MCU from the clocking cycles in the next version of these wireless clocks. Utilizing a dedicated integrated circuit (IC), such as a real-time clock chip, to control the internal counting and time tracking, as well as stabilizing the crystal temperature, could help to eliminate the drifting found internal to these units.

The real issue with the synchronization between these devices does not come from drift, however. Addressing question two on the list above shows that the drift rate is much less than the final synchronization error. Measurements utilizing the logic analyzer to track the Xbee communications showed what was primarily responsible for the synchronization error (Figure 4.3). In the particular sample taken in this figure, the time it took the master to transmit to the slave was approximately 7.5ms while the time it took the slave to transmit to the master was approximately 11ms. Error in the network delay comes from estimating that the round trip network delay is isotropic (meaning equal network delays in both directions). Over time, the delays in the Xbee network vary and do not remain constant. Thus, during each synchronization cycle, errors when measuring the Xbee link delays will result, and a final offset will emerge.

The best way to eliminate this offset would be to utilize a simpler setup in hardware with basic serial transmit/receive modules. The reason Xbees were utilized for this implementation was due to their built-in network management features. They can be configured to operate in a broadcast mode, or they can be configured in real-time to target a specific receiver. Utilizing the basic module hardware would require further programming to implement this function on the connected MCUs, but would provide a more accurate network delay, equal almost entirely to the separation distance between the units. Utilizing

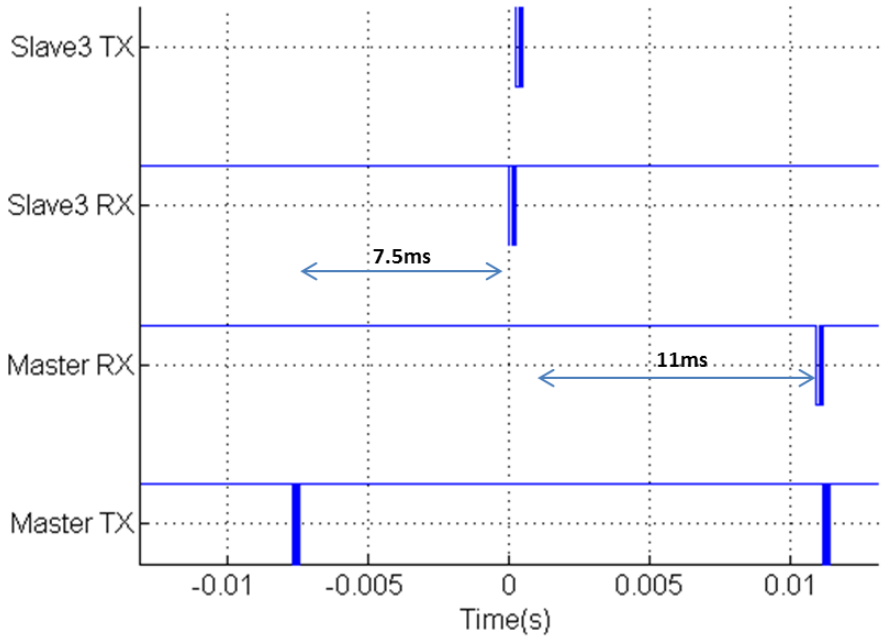


Figure 4.3: Logic analyzer measurement of the communication between the master and a slave clock unit showing the non-isotropic network delays sourcing the final synchronization errors

these modifications, final wireless synchronization accuracies could be achieved down to sub 500 microsecond ranges.

Implementation of these devices was not accomplished during this thesis research. The main reason was due in fact to the 4ms accuracy of the wireless clocks. With this error, a total of 8ms of data would need to be captured and processed for every network capture. This significantly degrades the refresh rate of the navigation network. This long capture length conflicted with the *tcpip* command functions. Since *tcpip* required large amounts of memory for buffers, these long captures could not be accomplished on the 32-bit computers used. These memory issues made it impractical for immediate implementation of the wireless clocks into the current version of the navigation network.

4.2 Indoor Multipath Analysis

Since the actual correlations outside of the simulation world are not perfect impulses, two different tests were conducted to determine the impacts that operating indoors will have on the final correlations. Both utilized the configuration seen in Chapter 3, where transmission was performed through building walls. The first test was conducted early on in experimentation, and consisted solely of a single trigger capture per antenna position. The reason for utilizing a single trigger was that initial experimentation averaging around 3 to 5 triggers showed little improvement over the single trigger capture case. Thus, it was decided to capture only a single trigger at a time during all future experiments to help improve the refresh rate of the system between each run at the navigation algorithm. This also aided in memory management issues in MATLAB. Post-analysis of the results from this test and others showed that a larger number of triggers (i.e. on the order of 10 or greater, dependent upon the RF environment) averaged over time would significantly aid in the stability of the correlation results. This number was much greater than the original estimations, and thus was not attempted early on. Unfortunately, the determination that many more triggers needed to be utilized during every capture was made late in the research process, after the main navigation captures took place. These captures could still be averaged over time, however, producing data that is still useful for analysis of the system navigation performance. The second indoor multipath analysis was performed with a large number of captures per antenna placement location, which were averaged after the fact.

The single trigger testing results are pictured in Figure 4.4. Note that the correlations show multiple returns in this indoor environment as expected. As the receive antenna was moved deeper into the room, lower power returns were seen in the direct path peaks following the basic phenomenon of EM propagations. The second peak group of returns corresponded directly to the length of the room, and each of the smaller peaks within each large group correspond to the width of the hallway in which the transmit antenna was

located. This shows that the transmitted energy is in fact reflecting off each wall, as well as transmitting through these walls and reflecting within the hallway.

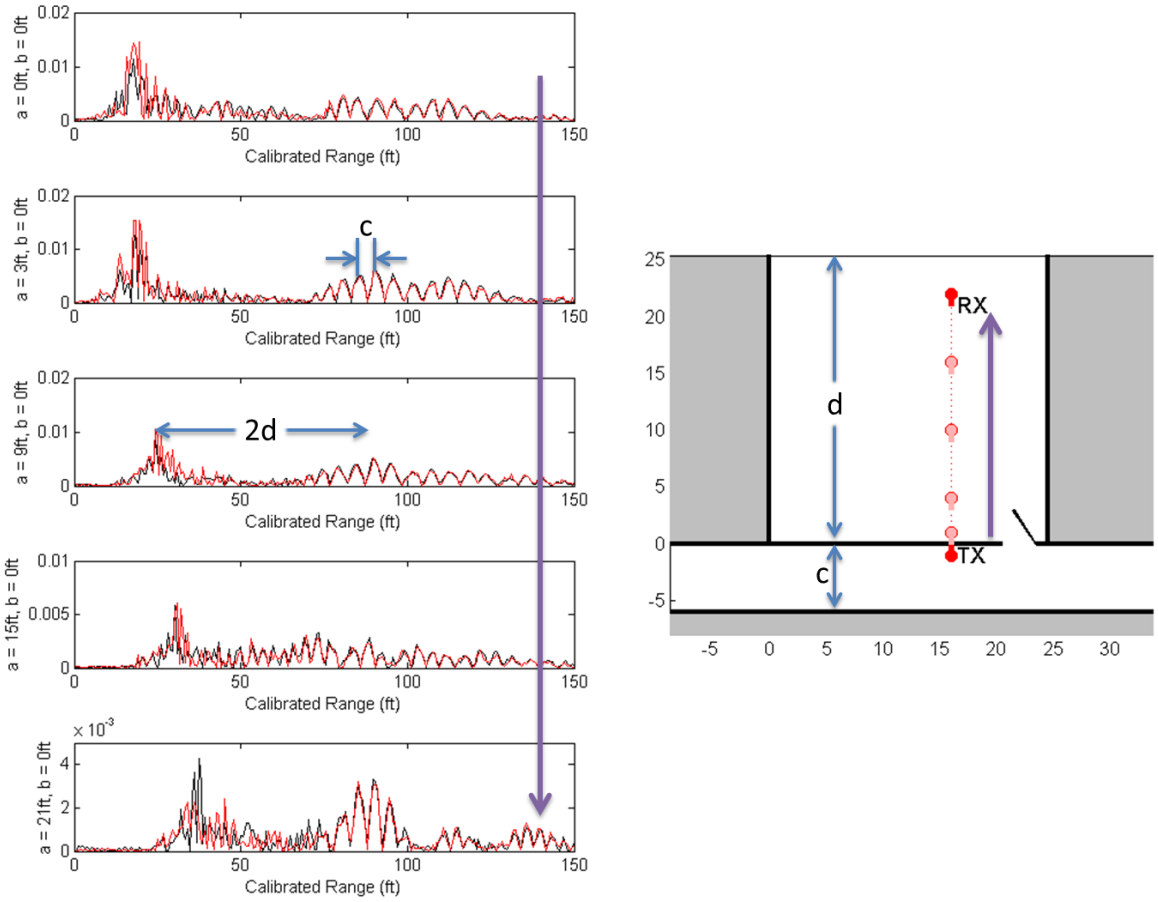


Figure 4.4: Diagram of the final single trigger correlation results for each range measurement taken annotating the reflections visualized and their corresponding ranges

Results were not heavily dependent upon the operation polarization when operating through walls. This shows that the reflections from the vertical surfaces (mainly the building walls) were dominant in comparison to the returns from the floors and ceiling of the room, which would be much more polarization dependent. Reflections from changes in the constitutive parameters of the propagation medium dominated the multipath effects.

Note that these measurements were taken with antennas aimed normal to these building walls. This orientation produces a worst-case scenario, where multiple returns will be perfectly reflected and captured on the receive antenna. By orienting the antennas at an oblique angle to the walls, the multiple reflections can be more easily mitigated, however, the direct path field strength will be much less than transmission levels when normal to the walls. At oblique incidence angles, more energy is reflected off the outside wall than energy that makes it into the room. Further analysis of this trade-off would be beneficial in the future, when transmitting through walls to perform this indoor navigation function.

Finally, note that in all the measurement cases even after calibrating out all cable lengths and internal component delays, the direct path returns for the first measurements were not centered close to zero feet. The returns were expected to reside near zero due to the fact that the two antennas were touching opposite sides of a 6 inch thick brick wall and were much closer together than the calibrated ranges showed. This issue left many questions as to why the physical connection calibration could not correct for and provide accurate one-way range estimates. The one-way range experiments discussed next in this document served to investigate this error source and determine whether or not an external calibration routine was needed.

4.3 Indoor One-way Range Experiments

In an effort to further quantify the accuracy of one-way range measurements utilizing NoNET, and to locate the source of the calibration issue seen during the indoor multipath measurements, one-way range measurements were taken. Two NoNET nodes were setup and calibrated by directly connecting the transmit antenna lead to the receive antenna lead. Figure 4.5 shows the results from this measurement. The multipath environment is heavily apparent; however, a direct path peak still stands above the rest of the returns. With a quick geometric analysis of this measurement configuration, it is estimated that in these measurements, the ground bounce reflection is the larger peak following the main return.

This equates to approximately 10ft of lag behind the primary return, due to both antennas being at an elevated position.

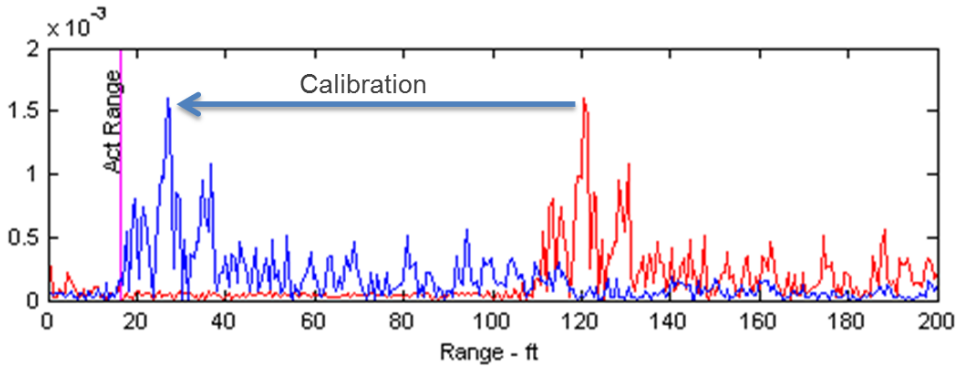


Figure 4.5: Correlation results from one-way range experimentation both uncalibrated and post-calibration showing peak offset from actual range

Interestingly enough, the direct return peak, though visible, was not calibrated properly. One would expect this peak to fall at the actual target range. For these measurements, this error was consistent (approximately 12ft at the speed of light). One consistent result, however, was that the calibrations brought the base of these peaks into alignment with the actual range every time. Since the calibration method utilized accounts for everything except the antennas, the logical conclusion is that the antennas must be the root cause for this calibration offset. For the antennas to induce this phenomenon, the error produced by each must be approximately 6ft obtained by dividing the total error in two, since there are two antennas in the RF pathway. Is it possible that these small antennas have an electrical length of 6ft?

To analyze this question, the NoNET LPAs were connected to a Time-domain reflectometer (TDR). This instrument analyzes reflection parameters in the time domain to estimate electrical resistance along a circuit pathway. Close visual inspection of the LPAs show the feed SMA connector on the front end connecting the central conductor of

the feed coaxial cable to one face of the double-sided circuit board. The shield or ground connects to the other face of the circuit board. Following the traces shows that at the tail end of the antenna, the sides of the circuit board are shorted together. Thus, one would expect the TDR to show some sort of taper from 50Ω to 0Ω over a length approximately equal to the antenna length.

First, a coaxial cable was connected to the TDR and left open on the opposite end. Here a marker can be placed on the time lag corresponding to where the resistance shifts from 50Ω to ∞Ω, or the end of the coaxial cable. Then the LPA was connected and a second marker was placed at the spot where the antenna approached 0Ω, or where a short exists on the antenna. Figure 4.6 shows the results from the TDR measurement.

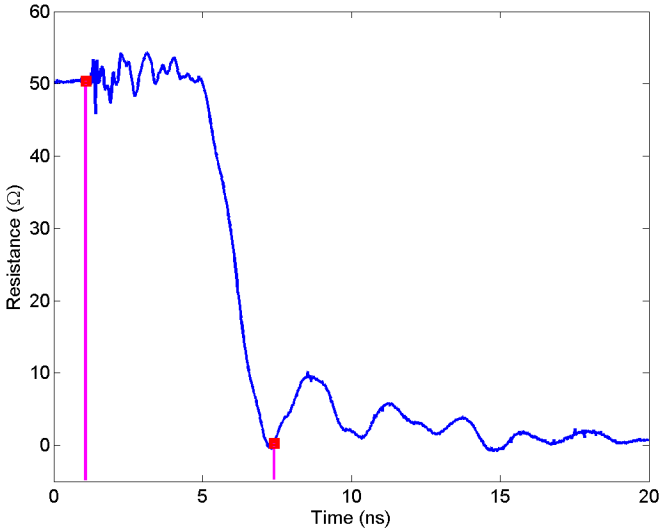


Figure 4.6: TDR measurement results of the LPAs showing a propagation delay of 6.3ns corresponding to an electrical length of 6.2ft

The difference between the markers is 6.3ns in time corresponding to an electrical length of the LPAs at approximately 6.2ft. Thus the offset seen in the one-way measurements is in fact caused by the antennas. The pre-response seen before the main

peak in the correlation is caused by higher frequency radiation being emitted before the lower frequency radiation. The shorter lengths of etched dipole traces on the circuit board are closer to the antenna feeds, and thus would begin radiating before the energy reaches the longer trace lengths. The lower frequency energy radiates at up to a 6ns delay. Thus, the LPAs are producing a Gaussian-like correlation response in the time domain. This effect inherently degrades the final range resolution achievable with NTR.

This effect is produced by the phase center movement with respect to frequency in the LPAs. Ludwig was able to simulate this movement in the phase center as frequency changed during his research. From 300MHz to 900MHz, he estimated via simulation that the phase center moved approximately 7ft in electrical length corresponding to the long TDR distance measured in this research [9].

To account for this lag, a calibration scheme was implemented in the NoNET navigation software, whereby a user places the receiver at a known location and performs a capture. Since all distances between nodes are known during calibration, the difference between the expected correlation peak locations and the actual peak locations can be calculated and used as a correction factor to adjust the measured psuedoranges before navigation algorithm processing.

4.4 Indoor Navigation Experiments

Testing of the navigation performance of the NoNET occurred in multiple locations, each with a different purpose. Testing in the AFIT ANT Center's lab was conducted on two separate occasions. This lab space contained a smaller open area with metal lined walls close to each of the transmitters. The results of this testing should produce a "worst case" scenario for multipath with larger emission reflection magnitudes than a standard brick or other interior wall.

4.4.1 AFIT ANT Center Lab Experimentation.

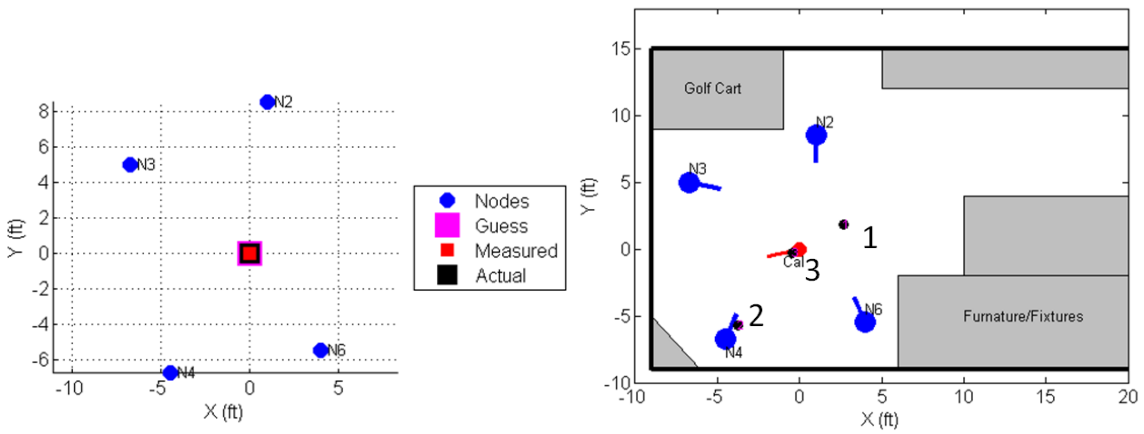
Stepping through one navigation iteration process from the ANT Center lab experimentation shows that the system is capable of locating a target. The test results provided an opportunity for analysis of the failed location attempts and their root causes.

Note that any analysis on the performance of the system via the final location estimations should also include a simulation of the navigation algorithm. In each case, since the algorithm used to calculate the final location is linearized, one must ensure that the correct guess was utilized in order to properly determine the end location. In other words, if a guess which was too far away from the actual receiver location is used in the algorithm, the final location result may still diverge, even if the NoNET measures the pseudoranges perfectly. Figure 4.7a is the result of the algorithm simulation for this test and configuration. Thus, if the NoNET correctly measures the ranges between each pair of nodes, the result should land very close to the origin.

Figure 4.7b diagrams the results of this navigation capture. A calibration was run with the receiver at the origin, which should have eliminated any correlation lag error due to the antennas or trigger error. The global captures were then conducted to ensure the system would then relocate the receiver at the origin as any error should have been calibrated out.

As seen in Figure 4.7b, it took three attempts for the system to localize the receiver. The question then becomes, what happened during the first two failed attempts? Analysis of the correlations utilized to determine the pseudoranges shows the error source. Figure 4.8 shows the correlations of two captures between the transmit waveform of Node 2 and the central node's receive waveform: capture 1, which failed to localize the receiver, and capture 3 which succeeded.

A large amount of information can be extracted from this particular set of correlations. Analysis will be conducted in the following areas: analysis of the variation between each measurement, the double peak seen in measurement 3 and calibration impacts, wall



(a) Simulation (b) Actual

Figure 4.7: Results of Setup 1 Iteration 1 navigation testing in the ANT Center's Lab, black circles correspond to the location outputs from the navigation algorithm

reflections and other multipath effects, effects from the LPAs, ADC truncation of the transmit waveforms, and the digital attenuation selections.

4.4.1.1 Ground Bounce Effects.

An interesting double peak (near the direct path peak) is seen in the correlation from capture 3 in Figure 4.8. For this particular measurement, the range difference between these two back to back peaks is approximately 3ft. A quick look at the geometry of this particular measurement instance shows that the ground bounce path is approximately 3.3ft of added distance from the direct path (Figure 4.9). Thus, the second peak is attributed to the ground reflection.

Note that the maximum peak is actually the ground reflection and not the direct path peak. Although 3ft of error in this pseudorange measurement existed, the receiver was still located properly due to the calibration. This extra 3ft was calibrated out since nothing was moved between that calibration and this set of measurements. Again, without the calibration, averaging aids in this particular problem, because, statistically, the direct path

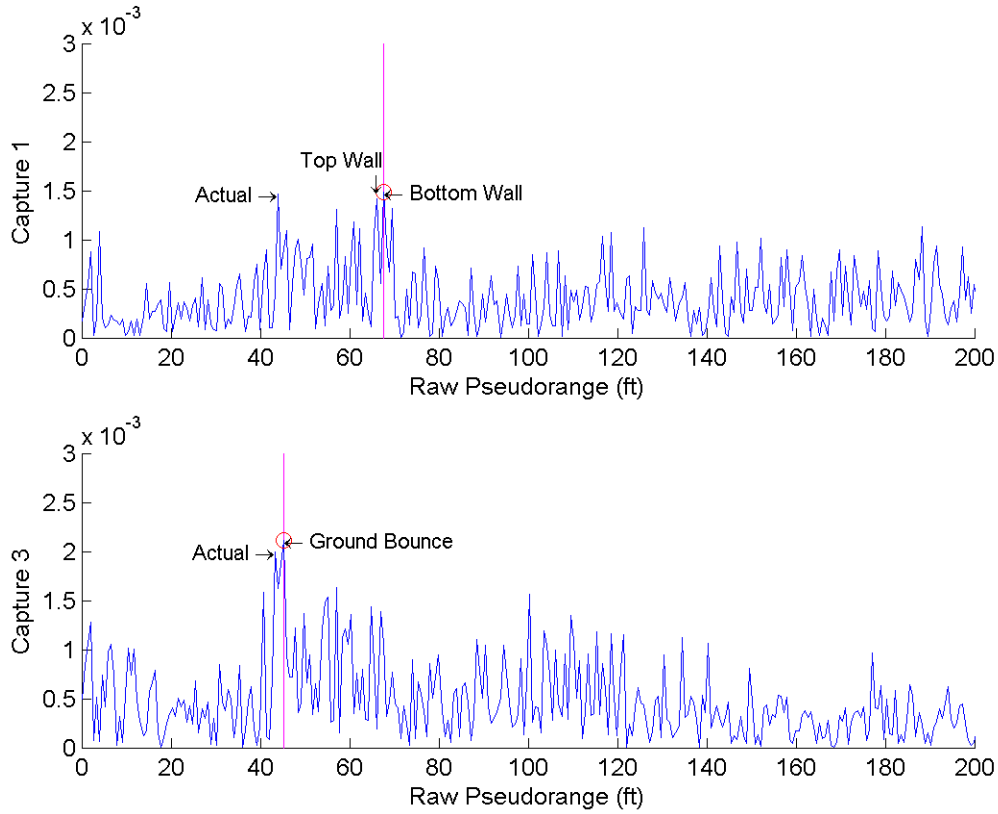


Figure 4.8: The raw correlations of Node 2's transmit waveform to the Central receive waveform showing the system utilizing the wrong maximum peak for pseudoranges during the first capture and showing the variation in correlation over time

response will be greater in magnitude to any response that has reflected off of another surface.

4.4.1.2 Wall Reflections and Other Multipath.

Quick measurements of the path lengths for reflections off both the top and bottom walls in Figure 4.7b shows the late time responses seen in capture 1 (Figure 4.8) are in fact from these multipath returns. In the case of this specific capture, these returns exceeded the levels of the direct path. With the walls in this particular testing area being metal, larger

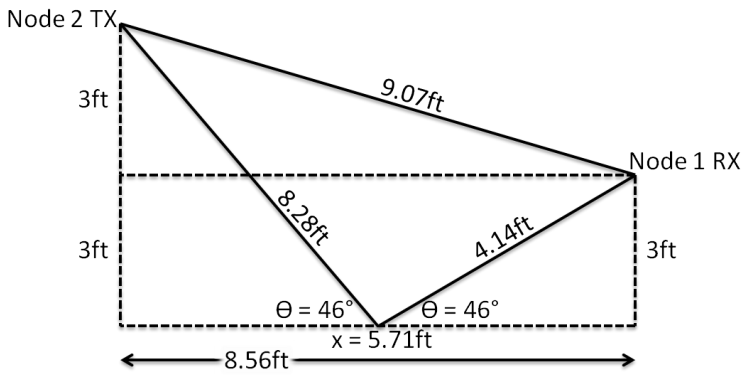


Figure 4.9: Diagram of the measurement geometry used to calculate an approximate ground reflection propagation path length

returns are expected, which again, could be mitigated via averaging over time, as the direct path return remains statistically larger in magnitude.

4.4.1.3 LPA Impacts.

As seen in the LPA analysis from Section 4.3, these antennas tend to provide what one could refer to as an early-time response in some situations. These particular antennas are constructed of a series of simple dipole radiating elements along their length. These elements start out smaller in size near the feed point (providing radiation at the upper frequencies) and taper up to larger sizes (providing radiation toward the lower frequencies). Due to the large electrical length seen in Figure 4.6, some of the higher frequency RF energy is radiated into the environment at an earlier time than the rest. Thus, some lower magnitude responses are captured at this earlier time lag, producing correlation peaks such as what is seen in the response from capture 3 (figure 4.8). Essentially, the phase center of the LPAs shift away from the feed point with a decrease in frequency. A quick test utilizing a set of captures taken with two antennas pointed at one another verifies this phenomenon. Figure 4.10 shows the correlation peak results from applying a sliding window to the

transmit and receive waveforms. As expected, as the frequency increases, the correlation peak moves to a shorter range.

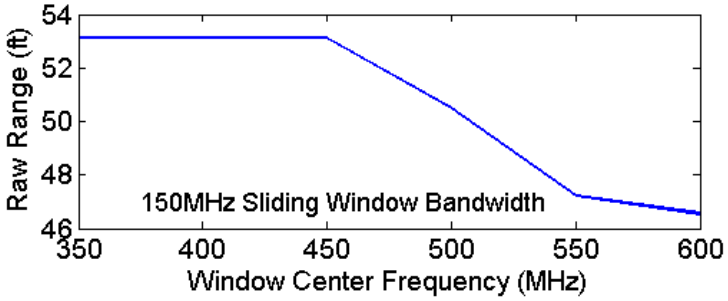


Figure 4.10: Impact on the pseudorange measurements from applying a sliding window in the frequency domain to the transmit and receive waveforms

With two LPAs pointed directly at one another, this effect is doubled, thus producing the roughly 12ft error post-calibration (Figure 4.5) from the roughly 6ft electrical length of each antenna. Note, however that if the receiving antenna were oriented in the same direction in line to the transmitting antenna, this effect should be eliminated. Thus, antenna geometries, when utilizing the LPAs, greatly impacts each response. In these navigation scenarios, where each transmitter will have a different orientation to the receiver, as well as to every other transmitter, these shifts due to the long electrical length of the antennas could prove problematic to final results. This produces large errors on the order of 6ft, which would not be correctable unless antenna orientations were known apriory.

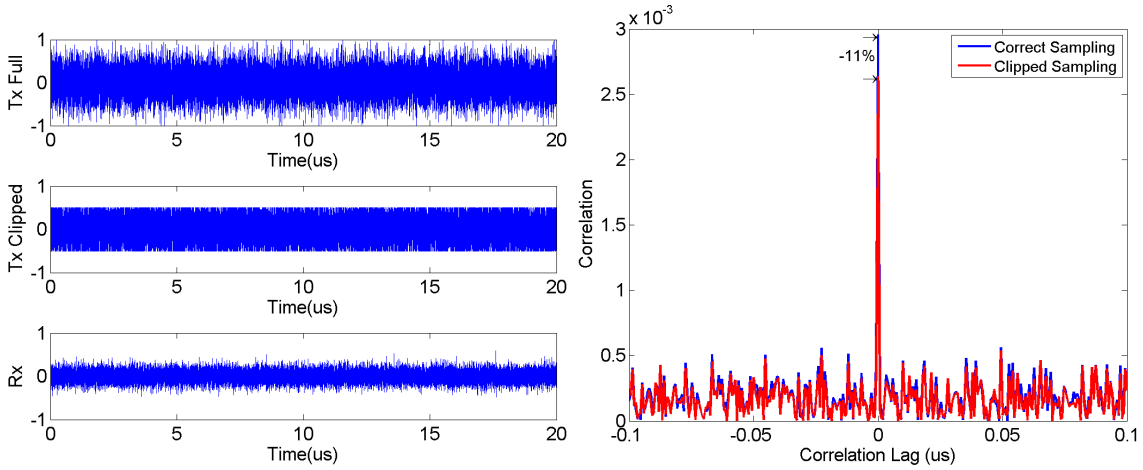
Another effect the LPAs will have on the system is the spectral coloring of the transmit and receive waveforms. These antennas do not have a flat response in frequency, and thus, will act as a filter in series with the RF pathways of NTR. Also note that the sampling conducted of the transmitted waveform by the ADC is performed in advance of the antennas, and does not take this filtering effect into account. In essence, some of the final signal power received by a remote NTR unit will be lost solely by utilizing these

specific antennas. Effects on the correlation between the transmit and receive signals will be a reduction in the peak amplitudes, further burying the signals into the noise floor of the environment.

4.4.1.4 Transmit Waveform Clipping.

One aspect of the measurements taken that was overlooked before detailed analysis was an error regarding the ADC dynamic range settings. Last minute comparisons of the transmit waveform sampled for this research to those waveforms captured during other work showed an error in the MATLAB code utilized to capture these receive waveforms. In order to set the ADC channel dynamic ranges to the proper voltage levels, each channel must be individually set. One line of code which set the receive channel dynamic range was found missing. This error forced the channel to operate from the desired $2V_{p-p}$ to the default of $1V_{p-p}$ thereby clipping the peaks in the noise waveforms.

To determine the extent to which clipping of the sampled transmit waveform affects the final correlations results, a quick simulation in MATLAB was run. This simulation utilized representative waveforms roughly equivalent in magnitude to those witnessed during testing with the NoNET. Figure 4.11a shows the waveforms utilized for correlation in this simulation. The top waveform is representative of a normal NTR ADC sampling of the transmit waveform. The middle signal is a simulation of the effects on this sampled transmit waveform when clipped. A correlation between each transmit waveform version and the simulated receive waveform can be seen in Figure 4.11b. These results show the impact of clipping the sampled transmit waveform clearly. Correlation results change in magnitude only. By approximating the NTR waveform captures as accurately as possible, the result of approximately 11 percent degradation can be considered accurate. Thus, impact from the ADC code error on the data only decreases the correlation levels, and affects both the multipath and the direct path results equally.



(a) Simulation Waveforms

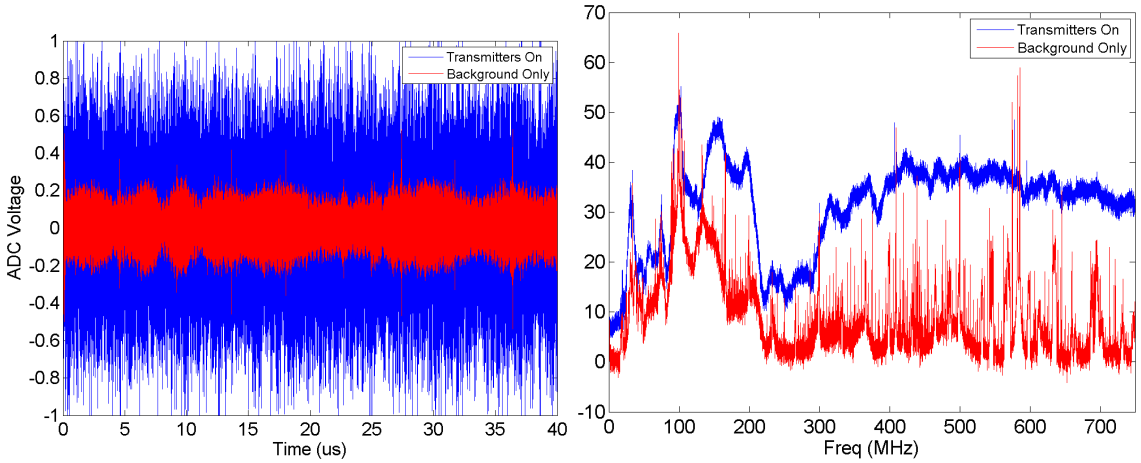
(b) Correlation Results

Figure 4.11: MATLAB simulation results analyzing the correlation impacts of clipping the sampled transmit waveforms

4.4.1.5 RFI Analysis.

When testing in an indoor environment with a large amount of other electronic equipment transmitting in the same location, analysis needed to be conducted to estimate the impacts this RFI had on the measurements. Receive captures were taken during testing while the entire network was shutdown, producing a true measurement of the background noise floor. Figure 4.12 shows the raw waveform captured and its spectrum obtained utilizing MATLAB's *fft* operation.

The results from this passive background capture demonstrate the effects RFI could have on the final performance of this system. First, raw voltage values from the ADC are on the order of $400mV_{p-p}$. When capturing the raw received waveforms with all NTR nodes on the network transmitting, the received voltage levels increase to approximately $1.5V_{p-p}$ providing an estimate of the signal to noise ratios during testing in this lab. Many strong deterministic signals are readily visible in both the time domain plot and the spectrum.



(a) ADC Capture Waveform (b) Normalized Receive Spectrum

Figure 4.12: Capture of the background RF environment during testing in the ANT Center's Lab

RFI impacts on the correlation results during navigation with NoNET will simply be the degradation of the correlation peaks in magnitude similar to the impacts of clipping the transmitted signal ADC sample.

4.4.1.6 Measurement Temporal Variations.

First, note that the room was perfectly stable with nothing having been moved or shifted in between these data captures. Utilizing the analog noise source for this process, as well as any background RFI, may have contributed to the instability of these measurements. A time-lapse video of the correlations was produced, and shows large correlation fluctuations over time with the direct path return dropping in and out. One solution to mitigation of these issues is to perform averaging over multiple captures, thereby smoothing out these temporal variations. Further indoor testing captured much larger datasets, in order to perform this function shown in Section 4.4.1.7. One drawback to capturing more triggers per capture, or averaging over each capture, is the limited available memory on-board the NoNET computers. The refresh rate of the system also degrades

significantly with added sets of data as a larger amount must be distributed across the network and must be correlated. The classic engineering tradeoff exists here. Navigation performance could be greatly improved if a more efficient method to send and receive data (other than *tcpip* links) were utilized, and a more efficient use of memory space within the MATLAB scrips were implemented.

4.4.1.7 Temporal Averaging Effects.

Due to the variations seen over time in the correlations, it was determined that many captures needed to be averaged to help improve the stability of the correlation peaks over time. This could be accomplished with a sliding average method, which would aid in maintaining the fluency of the correlations. Taking multiple triggers of data at a quick rate per global capture may not be as effective in mitigating RFI interactions. Allowing for longer periods of time between each sample averaged would be a more effective method. This assumes that the target or receiver is either stationary, or moving at a slow enough rate to accomplish this in real time.

Figure 4.13 is a plot of the pseudorange measurements for each capture taken in the ANT Center Lab. The averaging effectively removes the RFI from the measurements making the direct path correlation peak and all multipath effects stand out above the noise floor. As previously mentioned, the direct path should correspond to the correlation peak with the largest magnitude.

4.4.2 Kenney Hall Auditorium Experimentation.

Testing conducted in Kenney Hall Auditorium was aimed to help reduce some of the multipath effects seen in the AFIT ANT Center's lab space. With a more open environment having less clutter and angled walls preventing direct reflections, this area was ideal for producing cleaner indoor measurements (Figure 4.14).

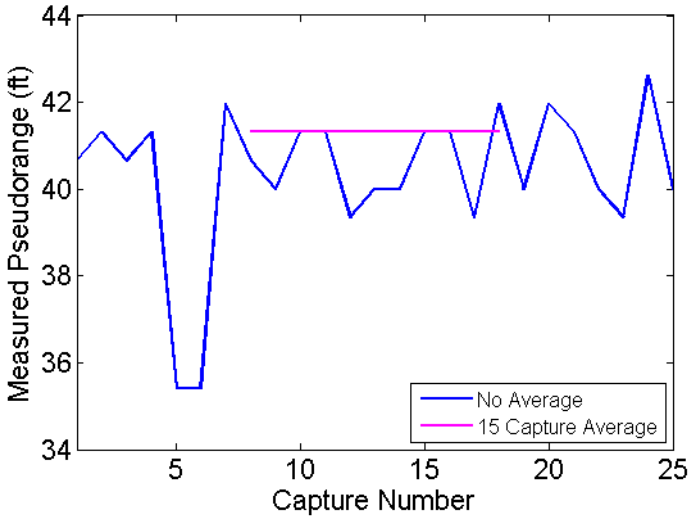


Figure 4.13: Impact analysis of the moving average implementation on psuedorange measurements

An aspect which plagued the final results of these measurements was the level of RFI present. Figure 4.15 shows the background captures from this experiment. Compared to the RFI during testing in the ANT Center's lab, the signal to noise ratio was less due to a couple of factors. The first is the slight increase in background noise, approximately 100mV. The second factor contributing to the lower signal to noise ratio levels is due to the increased separation distances between each of the transmitters. Comparing the ADC receive voltages between the Kenney Hall testing (Figure 4.15a) and the ANT Center testing (Figure 4.12b) demonstrates the effect on the signal voltage levels by separating each of the transmitters. The signal levels degrade significantly due to using a near-omni-directional antenna.

In Kenney Hall Auditorium, the network failed to locate the receiver. The reason for this failure was due mainly to poor correlation between the transmitters for clock error calculations. Specifically, Nodes 2 and 6, which were separated by the greatest distance, produced the most correlation error. Figure 4.16 shows the correlations between node 6

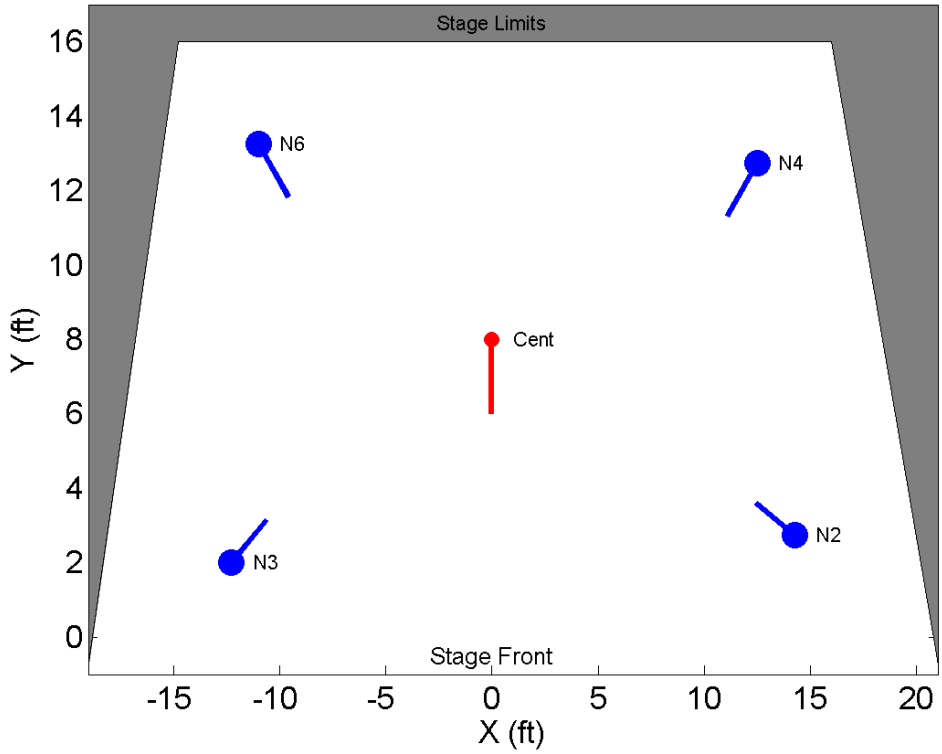
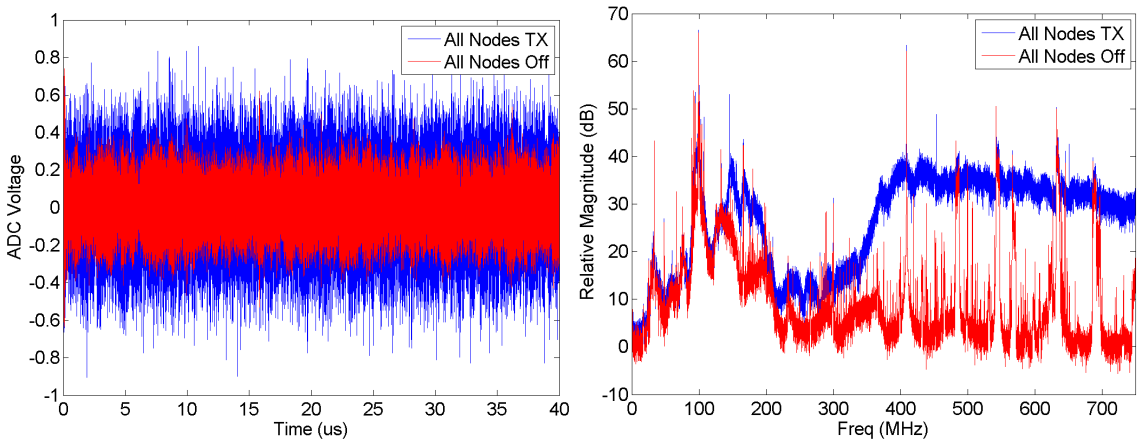


Figure 4.14: Setup configuration for testing in Kenney Hall Auditorium

and each of the other transmitters on the network. These correlations are averaged over 10 captures, which aides in extracting the direct path response from the noise floor. In this case, the correlation between Node 6 and 2 was unstable, and was averaged into the noise floor. Due to this result, the global clock error of each transmitter could not be extracted, and thus an accurate set of pseudoranges was not produced.

Time restrictions did not allow for repeat testing in this environment to produce a successful localization; however, the results still provide a vast amount of data for analysis. Analyzing the correlations which were stronger showed the significant reduction in multipath. In this case, reflections from the surrounding walls were eliminated mainly due to the slanting and increased separation of the walls. Kenney stage is constructed of metal, and thus the ground bounce returns still show up in the final correlations.



(a) ADC Capture Waveform

(b) Normalized Receive Spectrum

Figure 4.15: Capture of the background RF environment during testing in Kenney Hall Auditorium

4.5 Outdoor Navigation Experiment

Similar to the experimentation conducted in Kenney Hall Auditorium, the Outdoor captures were aimed to eliminate all multipath effects from the captures to obtain a true prediction of system performance. Ground bounce, however, is still an effect which was not eliminated in this environment. The two aspects of testing which prevented navigation success in Kenney Hall also plagued these outdoor measurements: RFI and node separations. The RFI during this testing was significantly greater in magnitude and was the primary contributer to navigation failure during this experiment. Correlation between the two transmitters separated by the greater distance was buried in the correlation noise floor.

One pair of nodes which were placed at a closer range to one another did provide useful correlation results. Only the direct path return and the ground bounce return appears above the noise floor (Figure 4.18).

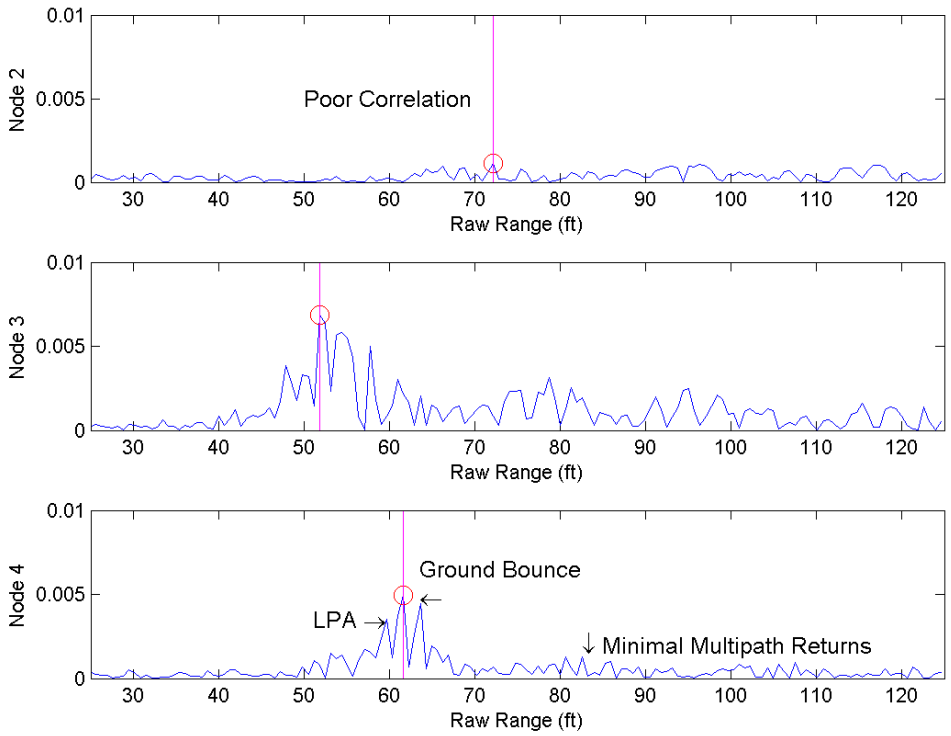
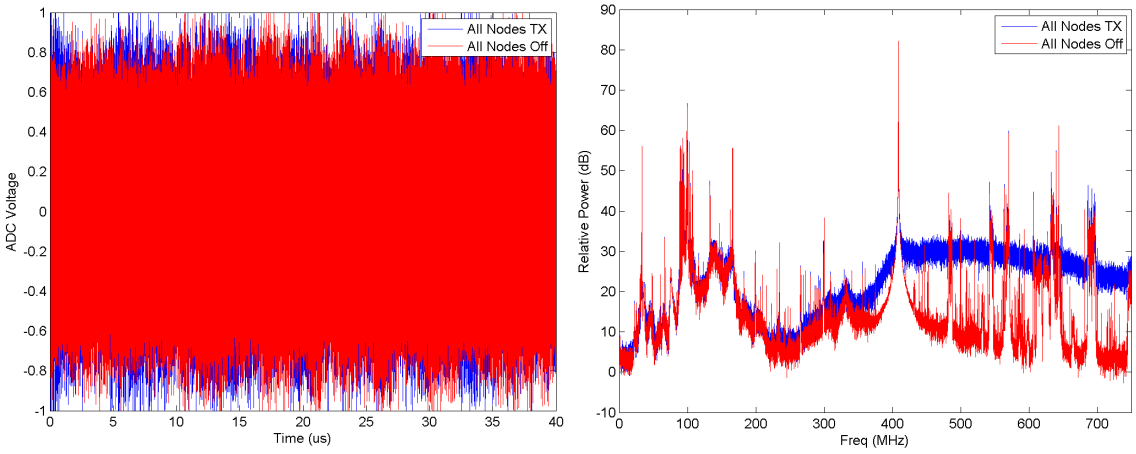


Figure 4.16: Node 6 correlations averaged over time to each transmitter on the network in Kenney Hall Auditorium



(a) ADC Capture Waveform

(b) Normalized Receive Spectrum

Figure 4.17: Capture of the background RF environment during outdoor testing

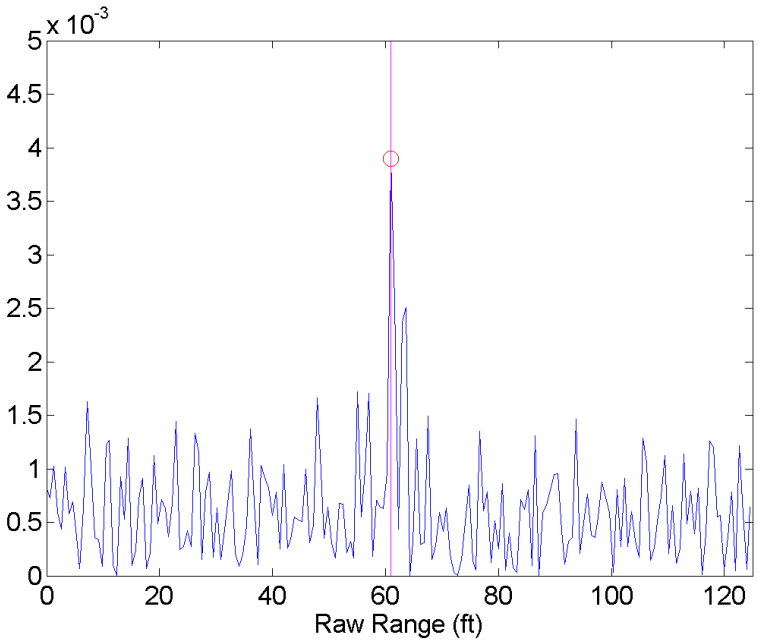


Figure 4.18: Receiver correlation with one transmitter's waveform during outdoor testing averaged over several captures

V. Conclusion

This chapter will provide a review of the work conducted. A review of the original objectives will be provided followed by an overview of the experimental results achieved and their contributions to the community at large. Finally, a glance at some of the projects future research could focus on will be outlined.

5.1 Objectives Review

The research objectives for this project centered around utilizing the AFIT NoNET for navigation in indoor and other cluttered environments. In order to perform this function, the following aspects required addressing:

- NoNET software development for navigation
- Demonstration and analysis of the NoNET's navigation performance
- Analysis of the effects from indoor multipath
- Development of a method for capture trigger synchronization and trigger error correction
- Development of a wireless trigger synchronization method

5.2 Results and Contributions

The results of this research demonstrated that the AFIT NoNET is capable of performing a navigation function in an indoor environment. It was shown that many factors degrade its performance. These include RFI, multipath, and other hardware-induced effects, such as antenna response. Analysis of the experimental data showed that each of these effects can be mitigated. The primary tool available is temporal averaging due to the noise waveform's orthogonality to any other RF waveform. By acquiring more

captures over time, signal to noise levels improve, and multipath effects become more easily extracted. In either case, the direct path responses will be the strongest and earliest return in a post-averaging scenario.

The other portion of the research objectives involved the global and wireless triggering of the NoNET captures. Methods for measuring and mitigating trigger error were constructed utilizing TDOA comparison techniques. A successful wireless triggering system was designed, constructed, and tested, which met the set requirements for NoNET. This research addressed many of the triggering issues, which prevented future development of not only navigation with the NoNET, but also operation of the traditional radar bistatic and multistatic modes.

5.3 Future Work

A great deal of work still exists in order to bring navigation utilizing the NoNET to a more practical and deployable level in its development. Many items which could significantly improve NoNET's performance in both navigation and in networked radar modes were discovered throughout the course of this research but were not implemented due to time restrictions. Suggested future work with NoNET includes:

- Implementation of a temporal moving average of the correlation results and outlier rejection

- Improvements to control over remote instances of MATLAB which require less memory allowing for longer and more frequent captures including transferring lower precision data across the tcpip links instead of the full double precision MATLAB utilizes

- Automatic adjustment of the digital receive attenuator gains and the ADC input dynamic ranges

- Parallelization of the processing and the capturing processes to allow for processing while the network is being setup for the next capture potentially utilizing MATLAB's *spmd()* command

- Utilization of antennas with less phase center movement in frequency

- Implementation of a real-time receive and processing capability similar to a communication receiver allowing for asynchronous captures

- Implementation of the template-based noise waveform Joshua Hardin's work developed, utilizing the DAC, which would eliminate the requirement to share the transmit data across the network

- Upgrades to the wireless clocks to utilize:

    · A temperature controlled oscillator

    · A Real-Time Clock Integrated Circuit instead of the MCU Timer/Counters

    · Communications modules which have a more isotropic network delay

Appendix A: Utilizing MATLAB's *tcpip* Function

This appendix will describe the MATLAB *tcpip* function and how it was utilized to pass commands and data between two remote instances of MATLAB. The *tcpip* function allows for the designation of the destination IP address, the port number, and the network role. The script below is an example of how *tcpip* can be utilized.

To setup a one-way link between two MATLAB instances, one instance must be designated a server and the other, the client. Thus, to obtain two-way communications, two *tcpip* objects need to be created. Note that since the same IP address is used for both of these links, a different port number must be specified for each link.

Using the command

```
tcpipObject = tcpip('localhost',5000,'NetworkRole','Server')
```

in MATLAB configures a *tcpip* object for use between two MATLAB instances on the same computer over port 5000 acting as a server. This object must then be opened using the *fopen()* command. The input and output buffers for each of these links must also be set to appropriate sizes for the data transfered using the MATLAB *set* command on the *tcpip* object. Open this link utilizing the MATLAB *fopen* command.

To write to and read from this object, utilize the fwrite and fread commands. One property of the *tcpip* object that is useful is the "BytesAvailable" property. Since the client must designate how much data to retrieve from the server and the precision level of the data, "BytesAvailable" can be utilized to grab the entire dataset.

A.1 Example Scripts

Added here are two scripts which demonstrate the usage of *tcpip* to perform a remote control function over a separate instance of MATLAB. To run this example, open two instances of MATLAB on the same computer. Launch the server script on one and the

client script on another. First the *tcpip* links will be established. The server script will then send a character string ”command” to the client. The client instance will attempt to execute this command.

A.1.1 Server Script.

```
tcpipServer = tcpip('localhost',5001,'NetworkRole','Server');
set(tcpipServer,'OutputBufferSize',10);
fopen(tcpipServer);

fwrite(tcpipServer,'a = 1','char');
```

A.1.2 Client Script.

```
tcpipClient = tcpip('localhost',5001,'NetworkRole','Client');
set(tcpipClient,'InputBufferSize',10);
fopen(tcpipClient);

ts = tic; % Timeout
while (toc(ts)<1000)
    try
        temp = fread(tcpipClient,tcpipClient.BytesAvailable,'char');
        eval(char(temp'));
    catch ME
    end
end
```

Appendix B: NoNET Navigation Software MATLAB Code

The code for navigation on NoNET starts with ensuring all subfolders are added to the path. From this point, launch the script entitled NoNETNavigation.m. This script reads the computer name of that particular computer and launches the appropriate software. For the receiver, NoNET-1, the NavGUI is launched. On the remote nodes, NoNET-2 through NoNET-6, the remotenodeNav.m script is launched. After TTL Card initialization on the remotes, they will be ready for connection. On the receive GUI, use the ”Detect” feature to determine which nodes are properly connected to the network. Then use ”Connect” to initialize the connections to all nodes. If this connection is lost or an error occurs, you must reset the entire script to reset the connections due to how *tcpip* functions. After this, the configurations can be input and updated which updates the configurations on the remote nodes as well. The begin button will start the process. On the first run of a MATLAB instance, the receiver will take about 2 minutes to initialize the TTL card. This is a bug in the system. For Joshua Hardin's work, this initialization is instantaneous. We believe that there are some conflicts in our code but could never determine the cause. At any time, press the stop button on the GUI to halt the capture process after the next iteration. The calibration feature was built in to account for one-way distance offsets due to the antennas. However, if good correlation is captured (using averaging later on), then the TDOA difference measurements will account for this in a crude sense.

B.1 Flow Diagrams

The main portions of the navigation software are diagrammed below. These are not all encompassing but aid in the understanding of how the captures were taken and processed.

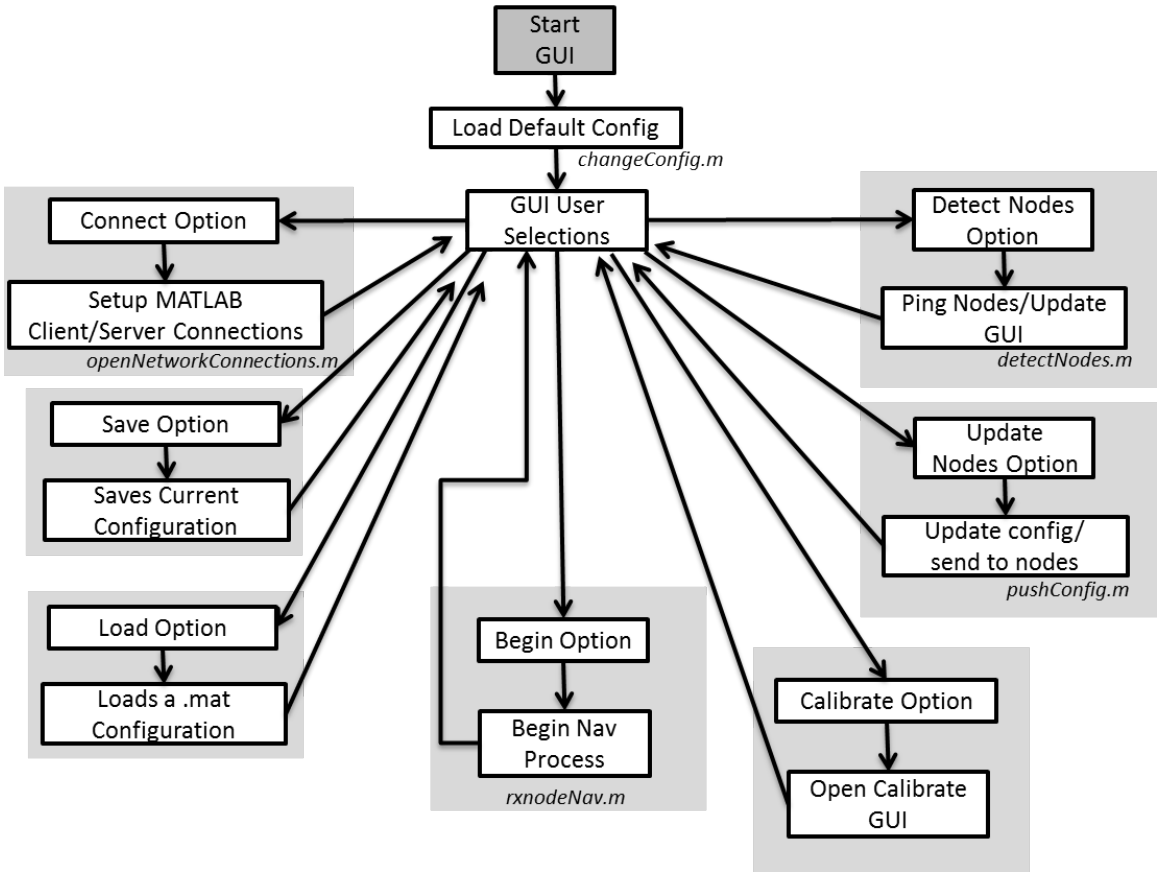


Figure B.1: Navigation GUI Software Flow on Receiver Node (NoNET-1)

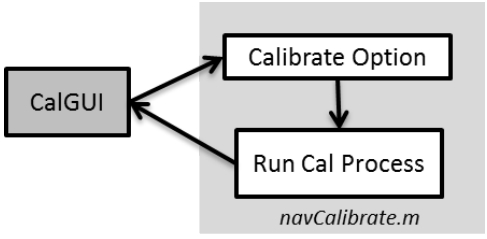


Figure B.2: Calibration GUI Software Flow on Receiver Node (NoNET-1)

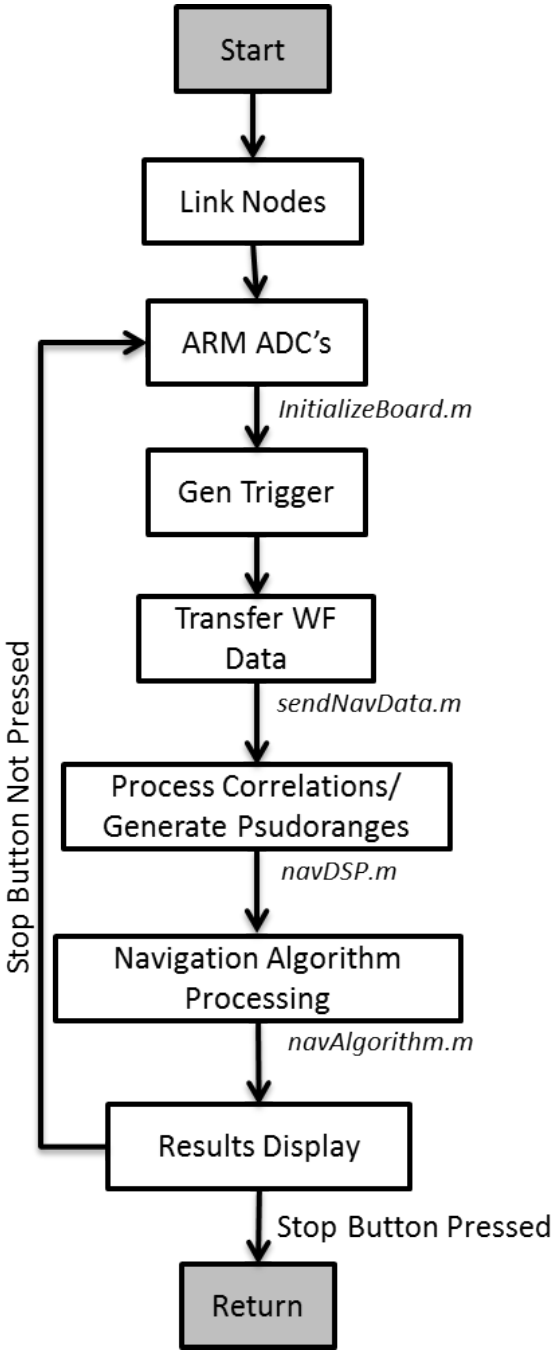


Figure B.3: rxnodeNav.m Software Flow - Main script for managing global captures on Receiver Node

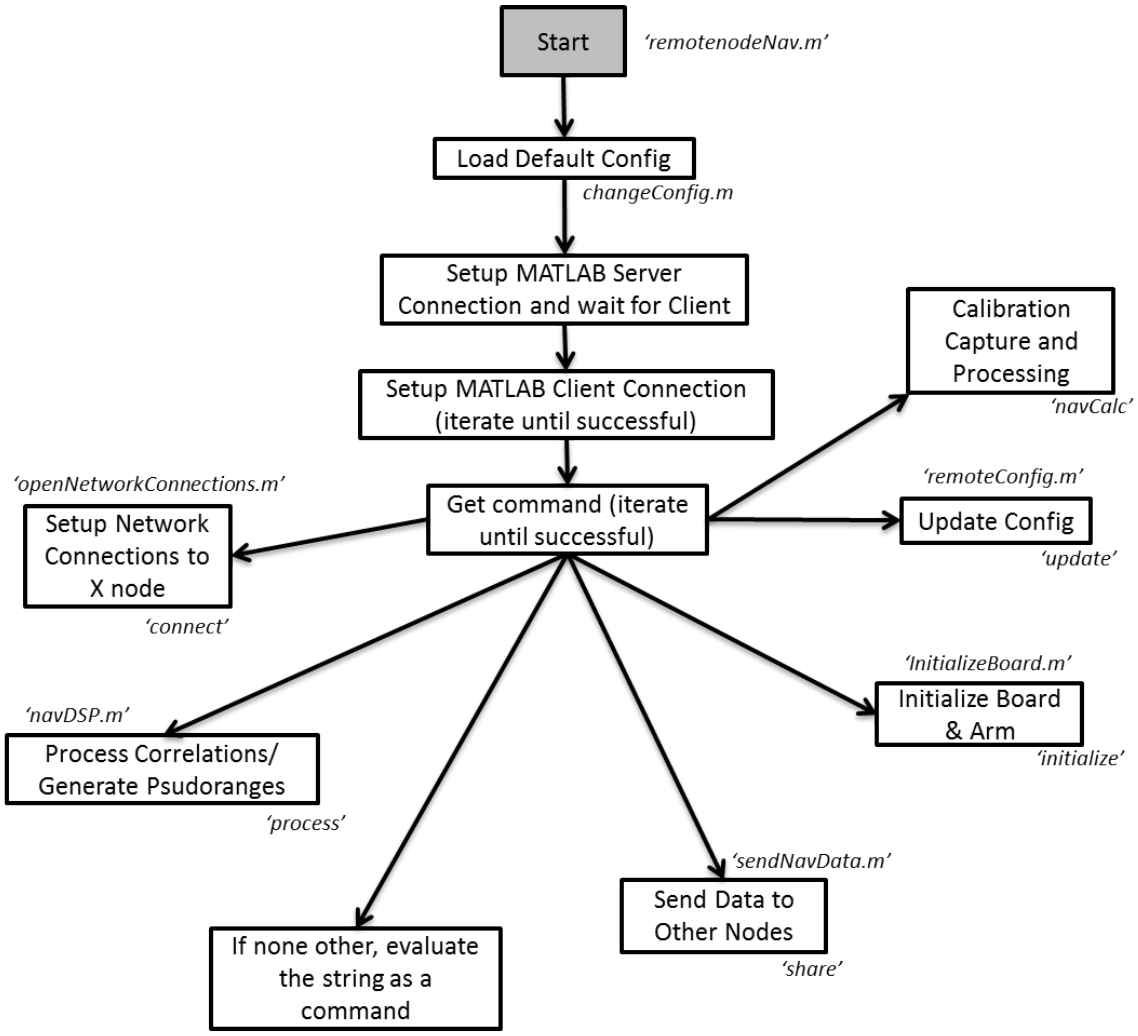


Figure B.4: remotenodeNav.m Software Flow - Main script ran on all Transmitters

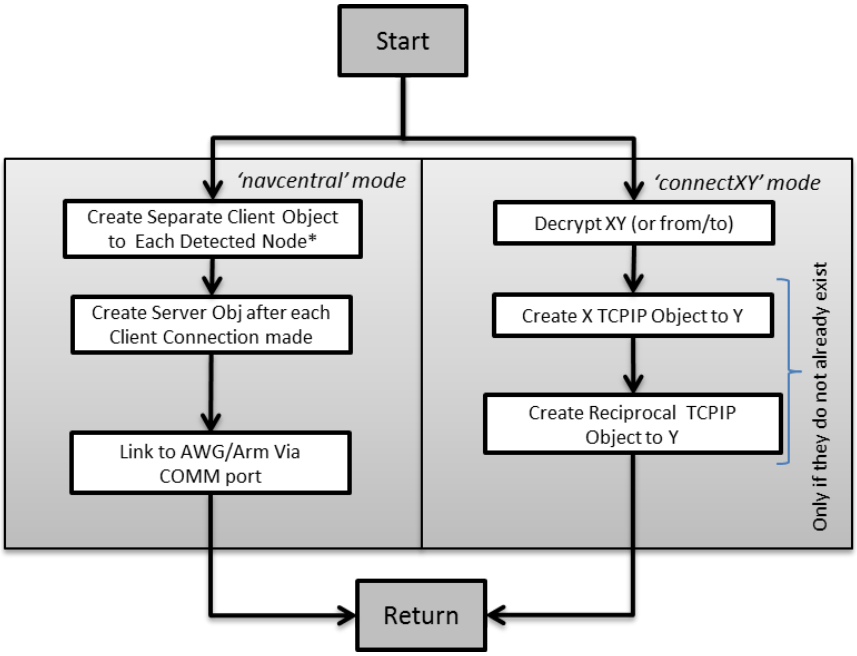


Figure B.5: openNetworkConnections.m Software Flow - Script linking all *tcpip* objects

Appendix C: NoNET Wireless Clock Detail

C.1 Master Software

```
/*
 * MasterClockv4.c
 *
 * Created: 11/29/2012 8:56:56 PM
 * Author: Lt Russell Wilson
 * NOTE: Modify delay.h Line 90 to '#define F_CPU 16000000UL'
 *
 * MCU: ATMEGA 168, 16MHz Crystal External, Xbee ZB
 */

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

#define F_CPU 16000000UL
#define USART_BAUDRATE 38400UL
#define BAUD_PRESCALE (((F_CPU/16UL)/USART_BAUDRATE)-1UL)

volatile uint16_t cmpVal = 31249; // The compare value relating to 2 seconds
volatile uint8_t trigState = 0; // Synthetic tracker of trigger state
volatile uint8_t rxNUM = 0; // How many rx interrupts have occured
volatile uint8_t readData; // Temp variable to store the RXed USART data
uint8_t slave; // Increment to step through each slave
uint32_t ii = 0; // increment for producing longer than millisecond delays
uint32_t timeOut = 0; // Used to measure time elapsed since Xbee Comm attempt
uint8_t complete = 0; // Used to determine successful Xbee communications
uint8_t txFlag = 0; //Flag to determine which rising edges to TX on
uint8_t NNout = 0; // Tracker as to when to TX to NoNET Xbee

int main (void){

  // Port configurations
  DDRB |= 0b00000101; // MCU Status LED output and Trigger Output
  PORTB = 0b00000101; // Initial LED activation on powerup
  ii = 0;
  while (ii <= 40) // Delay in initial MCU LED cycle
  {
      _delay_ms(50);
      ii++;
  }
  PORTB = 0; // Turn off MCU Status LED
  ii = 0;
  while (ii <= 20) // Delay in initial MCU LED toggle, est 1secs
  {
      _delay_ms(50);
```

```
    ii++;
}
ii = 0;// Counter for longer delays
while (ii <= 100) // Long wait to give XBee some boot time, est 5 sec
{
  _delay_ms(50);
  ii++;
}
// Blink MCU LED
PORTB |= (1 << 0); // Turn on MCU Status LED
ii = 0;
while (ii <= 20) // Delay in MCU LED toggle, est 1secs
{
    _delay_ms(50);
    ii++;
}
PORTB = (0 << 0); // Turn off MCU LED
ii = 0;
while (ii <= 20) // Delay in MCU LED toggle, est 1secs
{
    _delay_ms(50);
    ii++;
}
// Initial Configuration - Timer/Counter (T/C not started until initial sync)
TCCR1B |= (1 << WGM12); //Configure Timer 1 for CTC Mode (reset on compare)
TIMSK1 |= (1 << OCIE1B); // Enable compare B interrupt
TCCR1A |= (1 << COM1B0); // Enable Hard Pin toggle on compare, pin 16
sei(); // Enable Global Interrupts
OCR1B = cmpVal; // Set the compare values
OCR1A = cmpVal;
// Initial Configuration - USART
UBRR0H = (uint8_t)(BAUD_PRESCALE >> 8); // Set upper bits of baud rate
UBRR0L = (uint8_t)(BAUD_PRESCALE); // Set lower bits of baud rate
UCSR0B |= (1 << RXEN0) | (1 << TXEN0) | (1 << RXCIE0); // Enable RX and TX
  \\and RX interrupt
UCSR0C |= (1 << UCSZ00) | (1 << UCSZ01); //8Bit data, no parity, 1 stop bit
_delay_ms(50);
UDR0 = 'H'; // Send Hack command to all slaves to begin T/C's
TCCR1B |= ((1 << CS12)|(1 << CS10)); // Set T/C for ExtClock/1024 Prescale
PORTB |= (1 << 0); // Turn on MCU Status LED
ii = 0;
while (ii <= 4) // Delay in MCU LED toggle, est 1secs
{
    _delay_ms(50);
    ii++;
}
PORTB = (0 << 0); // Turn off MCU LED
while(1){ // Main loop
      NNout = 0; // Don't TX to X7 during sync
      performRemoteSync();
      // Delay between runs
      ii = 0;
```

```
        while (ii <= 1200) // Delay est 9 min
        {
            _delay_ms(50);
            ii++;
        }
  }
  return 0;
}

void enterCmdMode(void){
  rxNUM = 0; timeOut = 0; readData = 0;
  ii = 0;
  while (ii <= 50) // Delay, est 1.5secs
    {
    _delay_ms(50);
    ii++;
    }

  // With Xbee, enter command mode via '+++', then wait for an 'OK' back
  _delay_ms(1);
  UDR0 = '+'; // First +
  _delay_ms(1);
  UDR0 = '+'; // Second +
  _delay_ms(1);
  UDR0 = '+'; // Third +
  while (complete == 0)
  {
   while(rxNUM == 0 && timeOut <= 18000){_delay_ms(1); timeOut++;}
        // Wait for first RX character from Xbee (should be a 'O')
      timeOut = 0; // reset timeout in case multiple attempts
      if (readData == 'O' || readData == 'K' || readData == 0x0D) // 'O' RXed
      {complete = 1; break;} else {rxNUM == 0; readData = 0;} // or 'K' or '/r'
    if (complete == 0) // Another iteration of attempting Xbee Comm
    {
        ii = 0;
        while (ii <= 30) // Delay, est 1.5secs
        {
            _delay_ms(50);
            ii++;
        }
        UDR0 = '+'; // First +
          _delay_ms(1);
        UDR0 = '+'; // Second +
          _delay_ms(1);
        UDR0 = '+'; // Third +
          _delay_ms(1);
    }
  }
  complete = 0; rxNUM = 0;
}
```

```
void performRemoteSync(void){
  //Performing ping to each slave on pulse toggle
  uint8_t slaveResp;
  uint16_t tcVal;
  uint8_t numTemp;
  slave = 0;
  for (slave = 2; slave <=6; slave++)
  {
      enterCmdMode();
      complete = 0;
      _delay_ms(50);
      UDR0 = 'A'; _delay_ms(1); // Setting dest address, checks if slave is connected
      UDR0 = 'T'; _delay_ms(1);
      UDR0 = 'D'; _delay_ms(1);
      UDR0 = 'N'; _delay_ms(1);
      UDR0 = 'X'; _delay_ms(1);
      if (slave == 2){numTemp = 0x32;}
      if (slave == 3){numTemp = 0x33;}
      if (slave == 4){numTemp = 0x34;}
      if (slave == 5){numTemp = 0x35;}
      if (slave == 6){numTemp = 0x36;}
      UDR0 = (uint8_t)(numTemp); _delay_ms(1);
      readData = 0;
      UDR0 = 0x0D; _delay_ms(1);
      while (complete == 0) // wait for 'OK/r' or 'ERROR/r'
        {
        while(rxNUM == 0){_delay_us(1);} // Wait for RX character from Xbee
            if (readData == 'K') // was an 'O' or 'K' RXed
            {complete = 1; break;}
            if (readData == 'E' || readData == 'R'){complete == 2; break;}
        }
      if (complete == 1) // AKA a slave was detected
      {
        ii = 0;
        while (ii <= 20) // Quick Delay to exit command mode
        {
            _delay_ms(50);
            ii++;
        }

        complete = 0;
        txFlag = 1; // Setup to TX on next pulse
        while(complete == 0){_delay_us(1);}  // Wait for TX completion
        complete = 0;
      }
  }


  // Setup to TX to X7 (NoNET Computer)
      enterCmdMode();
```

```
        complete = 0;
        _delay_ms(50);
        UDR0 = 'A'; _delay_ms(1);
        UDR0 = 'T'; _delay_ms(1);
        UDR0 = 'D'; _delay_ms(1);
        UDR0 = 'N'; _delay_ms(1);
        UDR0 = 'X'; _delay_ms(1);
        numTemp = 0x37; // '7' ASCII
        UDR0 = (uint8_t)(numTemp); _delay_ms(1);
        readData = 0;
        UDR0 = 0x0D; _delay_ms(1);
        while (complete == 0) // wait for 'OK/r' or 'ERROR/r'
          {
          while(rxNUM == 0){_delay_us(1);} // Wait for RX character
              if (readData == 'K') // was an 'O' or 'K' RXed
              {complete = 1; break;}
              if (readData == 'E' || readData == 'R'){complete == 2; break;}
          }
        if (complete == 1) // AKA the 7th Xbee is linked into the network
          {
              NNout = 1; // Start TXing to X7 during sync
          } else { NNout = 0;}
}

ISR(TIMER1_COMPB_vect){ // Interrupt from T/C compare
   trigState ^= 1; // Tracking trigger state
   PORTB ^= (1 << 0); // MCU Status LED toggle
   if ((txFlag == 1) && (trigState == 1))// Send 'Hack' signal to slave
     {
        UDR0 = 'G'; txFlag = 0;
     }
   if ((NNout == 1) && (trigState == 1)) // Sending 'T' to X7 if exists
   {
       UDR0 = 'T';
   }
}

ISR(USART_RX_vect){ // Interrupt from USART RX Complete
  readData = (uint8_t)(UDR0); // Store register into memory
  if (readData == 'P') // If slave pinged, reply quickly
  {
      UDR0 = 'R';
      complete = 1;
  }
  rxNUM++; // Increment the rxNum for process tracking
}
```

C.2 Slave Software

```
/*
 * SlaveClockv4.c
 *
 * Created: 11/29/2012 8:56:26 PM
 *  Author: rdwilson
 * NOTE: Modify delay.h Line 90 to '#define F_CPU 16000000UL'
 *
 * MCU: ATMEGA 168, 16MHz Crystal External, Xbee ZB
 */

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

#define F_CPU 16000000UL
#define USART_BAUDRATE 38400UL
#define BAUD_PRESCALE (((F_CPU/16UL)/USART_BAUDRATE)-1UL)

uint16_t cmpVal = 31249; // All variables similar to the MasterClockv4.c file
uint8_t trigState;
uint8_t rxNUM = 0;
uint8_t readData = 0;
uint8_t complete;
uint8_t ii = 0;
uint16_t rxTCNT;
uint8_t adjTCNT = 0;

int main(void)
{
  // Port configurations
  DDRB |= 0b00000101; // MCU Status LED output & Pin Output
  PORTB = 0b00000101; // Initial LED activation on powerup

  while (ii <= 40) // Delay in initial MCU LED cycle
  {
      _delay_ms(50);
      ii++;
  }

  PORTB = 0; // Turn off MCU Status LED
  ii = 0;
  while (ii <= 20) // Delay in initial MCU LED toggle, est 1secs
  {
      _delay_ms(50);
      ii++;
  }

  // Initial Configuration - Timer/Counter (T/C not started until initial sync)
  TCCR1B |= (1 << WGM12); //Configure Timer 1 for CTC Mode (reset on compare)
```

```
  TIMSK1 |= (1 << OCIE1B); // Enable compare B interrupt
  TCCR1A |= (1 << COM1B0); // Enable Hard Pin toggle on compare, pin 16
  sei(); // Enable Global Interrupts
  OCR1B = cmpVal; // Set the compare value, enough room left for adjustments
  OCR1A = cmpVal;

  // Initial Configuration - USART
  UBRR0H = (uint8_t)(BAUD_PRESCALE >> 8); // Set upper bits of baud rate
  UBRR0L = (uint8_t)(BAUD_PRESCALE); // Set lower bits of baud rate
  UCSR0B |= (1 << RXEN0) | (1 << TXEN0) | (1 << RXCIE0); // Enable RX and TX
  UCSR0C |= (1 << UCSZ00) | (1 << UCSZ01); // 8Bit data, no parity, 1 stop bit
  PORTB |= (1 << 0);
  ii = 0;
  while (ii <= 8)
  {
      _delay_ms(50);
      ii++;
  }
  PORTB = (0 << 0);
  complete = 0; //Wait for 'H' Hack from master
  while(complete == 0){
      while(rxNUM == 0){_delay_us(1);};
      if (readData == 'H'){complete = 1; break;}
  }
  TCCR1B |= ((1 << CS12)|(1 << CS10)); // Set T/C for ExtClock/1024 Prescale

  PORTB |= (1 << 0);
  ii = 0;
  while (ii <= 8)
  {
      _delay_ms(50);
      ii++;
  }
  PORTB = (0 << 0);

    while(1)
    {
        tcUpdateAdj();
    }
}


void tcUpdateAdj(void){
    complete = 0; // Wait for 'G' from Master
    while(complete == 0){_delay_us(1);}
    complete = 0; // Ping 'P' sent to Master after 'G' RXed
    while(complete == 0){_delay_us(1);}
    // Calculate Network Delay
    rxTCNT /= 2;
    adjTCNT = 1;
    while (adjTCNT == 1){_delay_us(1);}
}
```

```
ISR(TIMER1_COMPB_vect){ // Interrupt from T/C compare
  PORTB ^= (1 << 0); // MCU Status LED toggle
  if (adjTCNT == 1)
  {
      OCR1B = cmpVal - rxTCNT;
      OCR1A = cmpVal - rxTCNT;
      adjTCNT = 0;
  } else {
      OCR1B = cmpVal;
      OCR1A = cmpVal;
  }
}

ISR(USART_RX_vect){ // Interrupt from USART RX Complete
  readData = (uint8_t)(UDR0); // Store register into memory
  if ((readData == 'G'))
  {
      TCNT1 = 0;
      UDR0 = 'P';
      complete = 1;
      PORTB ^= (1 << 0);
  }
  if (readData == 'R')
  {
      rxTCNT = TCNT1;
      complete = 1;
  }
  rxNUM++;
}
```

C.3 Hardware

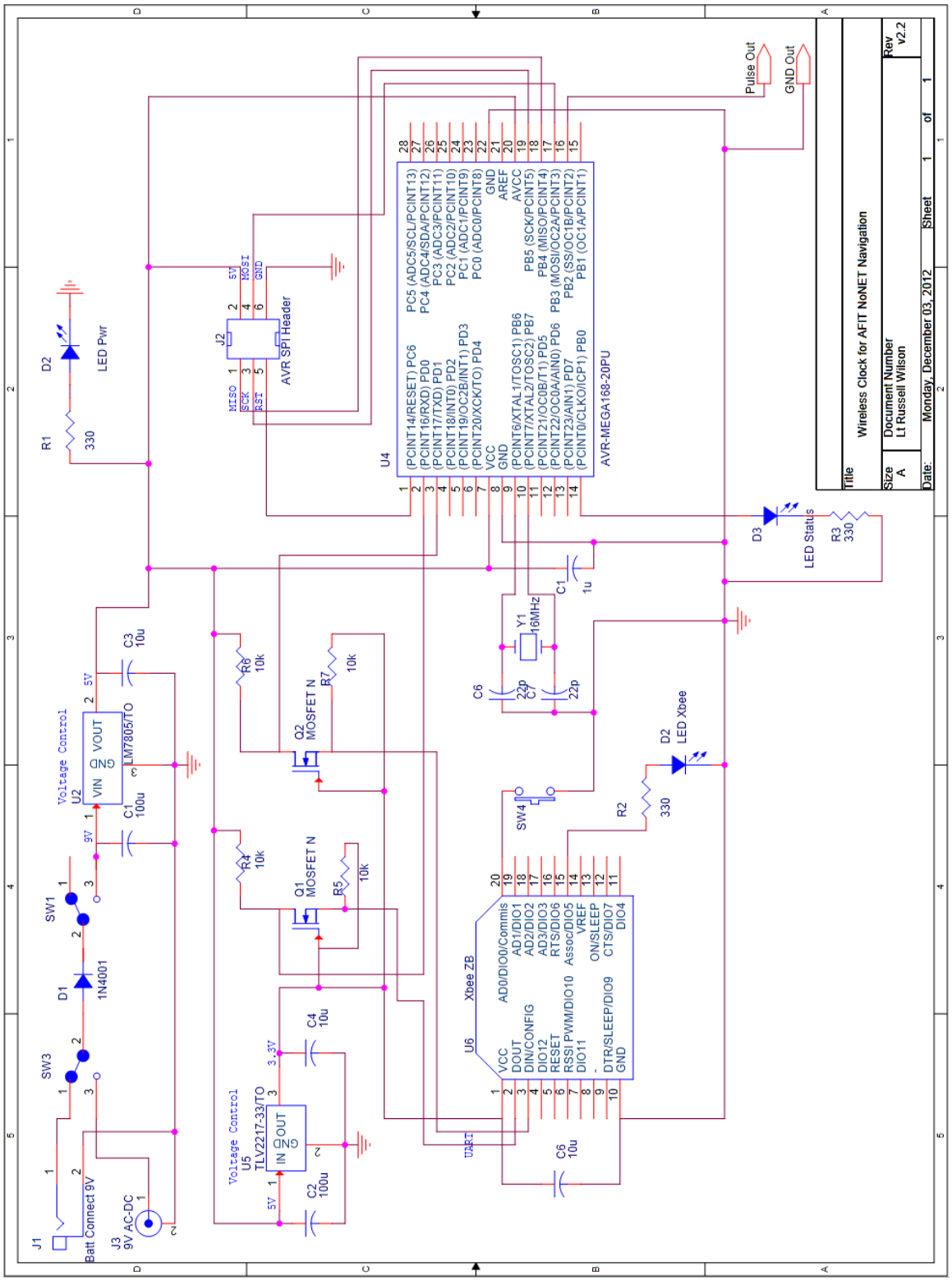


Figure C.1: Hardware schematic of the NoNET wireless clocks

Bibliography

[1] "IEEE Standard for Ultrawideband Radar Definitions", 2007. ID: 1.

[2] Balanis, C. A. *Advanced Engineering Electromagnetics*. Wiley, 1 edition, 1989.

[3] Barrett, T. W. "History of UltraWideBand (UWB) Radar & Communications:Pioneers and Innovators". *Progress In Electromagnetics Symposium 2000*. July 2000.

[4] Challamel, R., P. Tome, D. Harmer, and S. Beauregard. "Performance assessment of indoor location technologies". *Position, Location and Navigation Symposium, 2008 IEEE/ION*, 624 –632. may 2008.

[5] Hardin, J. *Information Encoding on a Pseudo-random Noise Radar Waveform*. Master's thesis, Air Force Institute of Technology, 2950 Hobson Way, WPAFB, OH, March 2013.

[6] Kaplan, Christopher J, Elliott D; Hegarty. *Understanding GPS: Principles and Applications*. Artech House, 2 edition, 2006.

[7] Lai, C. P. *Through Wall Surveillance Using Ultrawideband Random Noise Radar*. Ph.D. thesis, The Pennsylvania State University, 2007.

[8] Lievsay, J. R. *Simultaneous Range/Velocity Detection with an Ultra-Wideband Random Noise Radar Through Fully Digital Cross-Correlation in the Time Domain*. Master's thesis, Air Force Institute of Technology, 2950 Hobson Way, WPAFB, OH, March 2011.

[9] Ludwig, M. T. *UHF Antenna Design for AFIT Random Noise Radar*. Master's thesis, Air Force Institute of Technology, 2950 Hobson Way, WPAFB, OH, March 2012.

[10] Nelms, M. E. *Development And Evaluation Of A Multistatic Ultrawideband Random Noise Radar*. Master's thesis, Air Force Institute of Technology, 2950 Hobson Way, WPAFB, OH, March 2010.

[11] Richards, J. A.; Holm W. A., M. A.; Scheer. *Principles of Modern Radar: Basic Principles*, volume 1. Scitech, 2010.

[12] Schmitt, A. *Radar Imaging with a Network of Digital Noise Radar Systems*. Master's thesis, Air Force Institute of Technology, 2950 Hobson Way, WPAFB, OH, March 2009.

[13] Thorson, T. J. *Simultaneous Range-Velocity Processing and SNR Analysis of AFIT's Random Noise Radar*. Master's thesis, Air Force Institute of Technology, 2950 Hobson Way, WPAFB, OH, March 2012.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
21–03–2013	Master's Thesis	Oct 2011–Mar 2013

4. TITLE AND SUBTITLE

Adaptations and Analysis of the AFIT Noise Radar Network for Indoor Navigation

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Wilson IV, Russell D., Second Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-13-M-50

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Advanced Navigation Technology Center
2950 Hobson Way
WPAFB, OH 45433

10. SPONSOR/MONITOR'S ACRONYM(S)

ANT

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

After several years of development, the AFIT Noise Radar Network (NoNET) has proven to be an extremely versatile system for many standard radar functions. This pallet of capabilities includes through the wall target tracking capabilities due to its wide bandwidth and UHF operations. Utilizing White Gaussian Noise as its waveform, the NoNET can operate at much lower power levels than other comparable systems while remaining extremely covert. In an effort to explore new applications, the question arose could the NoNET provide a viable option for navigation capability in GPS denied and indoor environments? This research aims to provide proof of concept and demonstration of the navigation function execution with the NoNET in indoor, multipath-ridden environments. Results demonstrate that the NoNET is currently capable of locating a receiver to accuracies of approximately 1 foot. Multipath, background RF interference, and network timing were investigated and solutions to mitigate the limitations imposed by each were developed with the potential to significantly improve accuracy. Future upgrades to the current NoNET hardware package were also investigated in order to provide a near real-time, portable solution to navigation in GPS denied environments.

15. SUBJECT TERMS

Noise Radar, Indoor Navigation, UHF Radar, Ultra-wideband, GPS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Peter J. Collins (ENG)
U	U	U	UU	108	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636 x7256 peter.collins@afit.edu